Lecture Notes in Computer Science 4299

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Steve Renals Samy Bengio
Jonathan G. Fiscus (Eds.)

Machine Learning for Multimodal Interaction

Third International Workshop, MLMI 2006
Bethesda, MD, USA, May 1-4, 2006
Revised Selected Papers

Springer

Volume Editors

Steve Renals
University of Edinburgh
The Centre for Speech Technology Research
Edinburgh EH8 9LW, UK
E-mail: s.renals@ed.ac.uk

Samy Bengio
IDIAP Research Institute
CP 592, Rue du Simplon, 4, 1920 Martigny, Switzerland
E-mail: bengio@idiap.ch

Jonathan G. Fiscus
National Institute of Standards and Technology (NIST)
100 Bureau Drive, Gaithersburg, MD 20899-8940, USA
E-mail: jfiscus@nist.gov

Library of Congress Control Number: 2006938909

CR Subject Classification (1998): H.5.2-3, H.5, I.2.6, I.2.10, I.2, I.7, K.4, I.4

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-69267-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-69267-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11965152 06/3142 5 4 3 2 1 0

Preface

This book contains a selection of refereed papers presented at the 3rd Workshop on Machine Learning for Multimodal Interaction (MLMI 2006), held in Bethesda MD, USA during May 1–4, 2006.

The workshop was organized and sponsored jointly by the US National Institute for Standards and Technology (NIST), three projects supported by the European Commission (Information Society Technologies priority of the sixth Framework Programme)—the AMI and CHIL Integrated Projects, and the PASCAL Network of Excellence—and the Swiss National Science Foundation national research collaboration, IM2.

In addition to the main workshop, MLMI 2006 was co-located with the 4th NIST Meeting Recognition Workshop. This workshop was centered on the Rich Transcription 2006 Spring Meeting Recognition (RT-06) evaluation of speech technologies within the meeting domain. Building on the success of previous evaluations in this domain, the RT-06 evaluation continued evaluation tasks in the areas of speech-to-text, who-spoke-when, and speech activity detection.

The conference program featured invited talks, full papers (subject to careful peer review, by at least three reviewers), and posters (accepted on the basis of abstracts) covering a wide range of areas related to machine learning applied to multimodal interaction—and more specifically to multimodal meeting processing, as addressed by the various sponsoring projects. These areas included human–human communication modeling, speech and visual processing, multimodal processing, fusion and fission, human–computer interaction, and the modeling of discourse and dialog, with an emphasis on the application of machine learning. Out of the submitted full papers, about 50% were accepted for publication in the present volume, after authors had been invited to take review comments and conference feedback into account. The workshop featured invited talks from Roderick Murray-Smith (University of Glasgow), Tsuhan Chen (Carnegie Mellon University) and David McNeill (University of Chicago), and a special session on projects in the area of multimodal interaction including presentations on the VACE, CHIL and AMI projects.

Based on the successes of the first three MLMI workshops, and to strengthen and broaden the base of this workshop series, the MLMI standing committee was formed. The initial membership comprises Samy Bengio (IDIAP), Hervé Bourlard (IDIAP and EPFL), Tsuhan Chen (Carnegie Mellon University), John Garofolo (NIST), Mary Harper (Purdue University), Sharon Oviatt (Natural Interaction Systems), Steve Renals (Edinburgh University), Rainer Stiefelhagen (Universität Karlsruhe), and Alex Waibel (Carnegie Mellon University and Universität Karlsruhe). The committee will provide a permanent link across MLMI workshops. MLMI 2007, the fourth workshop in the series, will take place in

Brno, Czech Republic during June 28–30, 2007, directly after ACL–2007, which takes place in Prague.

Finally, we take this opportunity to thank our Programme Committee members, the sponsoring projects and funding agencies, and those responsible for the excellent management and organization of the workshop and the follow-up details resulting in the present book.

October 2006

Steve Renals
Samy Bengio
Jonathan Fiscus

Organization

Organizing Committee

Samy Bengio	IDIAP Research Institute
Hervé Bourlard	IDIAP Research Institute
Jonathan Fiscus (Co-chair)	NIST
John Garofolo	NIST
Steve Renals (Co-chair)	University of Edinburgh
Vincent Stanford (Demonstrations)	NIST
Alex Waibel (Special Sessions)	CMU and Universität Karlsruhe

Workshop Organization

Patrice Boulanger	NIST
Caroline Hastings	University of Edinburgh
Avril Heron	University of Edinburgh
Jonathan Kilgour	University of Edinburgh
Teresa Vicente	NIST

Program Committee

Marc Al-Hames	Munich University of Technology
Tilman Becker	DFKI
Jean Carletta	University of Edinburgh
Dan Ellis	Columbia University
Corinne Fredouille	University of Avignon
Thomas Hain	University of Sheffield
Mary Harper	Purdue University
Thomas Huang	University of Illinois
Alejandro Jaimes	Fuji Xerox
Samuel Kaski	University of Helsinki
Stephane Marchand-Maillat	University of Geneva
Nelson Morgan	ICSI
Andrei Popescu-Belis	University of Geneva
Mubarak Shah	University of Central Florida
Rainer Stiefelhagen	Universität Karlsruhe
Jean-Philippe Thiran	EPFL
Victor Tom	BAE Systems
Pierre Wellner	IDIAP Research Institute

Sponsoring Projects and Institutions

Institutions:

- US National Institute of Standards and Technology (NIST), http://www.nist.gov/speech/
- European Commission, through the Multimodal Interfaces objective of the Information Society Technologies (IST) priority of the sixth Framework Programme
- Swiss National Science Foundation, through the National Center of Competence in Research (NCCR) program

Projects:

- AMI, Augmented Multiparty Interaction, http://www.amiproject.org/
- CHIL, Computers in the Human Interaction Loop, http://chil.server.de/
- PASCAL, Pattern Analysis, Statistical Modeling and Computational Learning, http://www.pascal-network.org/
- IM2, Interactive Multimodal Information Management, http://www.im2.ch/

Table of Contents

MLMI'06

VI Speech and Audio Processing

VII NIST Meeting Recognition Evaluation

Model-Based, Multimodal Interaction in Document Browsing

Parisa Eslambolchilar[1] and Roderick Murray-Smith[1,2]

[1] Hamilton Institute, National University of Ireland, Maynooth, Co.Kildare, Ireland
parisa.eslambolchilar@nuim.ie
[2] Department of Computing Science, Glasgow University, Glasgow, Scotland
rod@dcs.gla.ac.uk

Abstract. In this paper we introduce a dynamic system approach to the design of multimodal interactive systems. We use an example where we support human behavior in browsing a document, by adapting the dynamics of navigation and the visual feedback (using a focus-in-context (F+C) method) to support the current inferred task. We also demonstrate non-speech audio feedback, based on a language model. We argue that to design interaction we need models of key aspects of the process, here for example, we need models for the dynamic system, language model and sonification. We show how the user's intention is coupled to the visualization technique via the dynamic model, and how the focus-in-context method couples details in context to audio samples via the language identification system. We present probabilistic audio feedback as an example of a multimodal approach to sensing different languages in a multilingual text. This general approach is well suited to mobile and wearable applications, and shared displays.

1 Introduction

In [1], McCullough writes about the need to simultaneously engage both a human's brain and hands, that media have to be dense enough to give the impression of a universe of possibilities. In this paper we present a continuous interaction, dynamic simulation approach which leads naturally to the sort of organic, rich interaction desired by McCullough. It also provides the potential for a solid, systematic way to develop future multimodal interaction systems.

We use tools to control, interact and operate on the physical objects rather than using our bare hands [2]. Instrumental Interaction [3] is an interaction model that operationalizes the computer-as-tool paradigm and extends human powers: a piece of technology, or applied intelligence for overcoming the limitations of the body and controlling information flow [1].

Continuous control is at the very heart of tool usage in the interaction between the human and computer as a tool [1]. It differs from discrete interaction in that it occurs over a period of time, in which there is an ongoing relevant exchange of information between user and system at a relatively high rate, somewhat akin to vision/audio/haptic interfaces which we may not model appropriately

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 1–12, 2006.

as a series of discrete events [4]. It is also closely related to the development of dynamic systems since in these systems we can control what we perceive and we are dependent on the display of feedback (either visual, audio or haptic) to help us pursue our potentially constantly changing goals. Furthermore, feedback may influence an uncertain user's actions as more information becomes available [5].

In order to address the behavioral issues early in the design stage, formal modeling techniques for real-time systems supported by powerful analysis tools could be considered and for calibration and refinement issues, a more general framework that can guide the modeling approach is needed.

In this paper, as an illustration of how this approach can support multimodal interaction, we use the example of browsing and sensing multilingual texts. Here the focus-in-context method and the adaptive dynamics are coupled with sonification, based on a probabilistic language model, which can be linked to a wide range of inputs and feedback/display mechanisms.

2 Continuous Interaction and Text Browsing

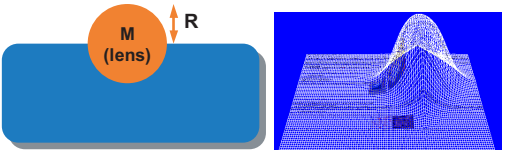
Our interaction model is an example of *continuous interaction* which means the user is in constant and tightly coupled interaction with the computing system over a period of time. Here, we use control theory as a formal framework for analysis and design of continuous interaction, multimodal feedback and overall system dynamics.

Focus-in-context methods are useful for displaying information in context and can be applied to various objects [6,7,8,9,10]. As our integrated system benefits from an Elastic Presentation Framework (EPF) [11], the presentation has an elastic nature. Elastic is a positive word that implies adjusting shape in a resilient manner, which means these materials can always revert to their original shape with ease. One popular way of describing a conceptual model [12] in terms of interaction metaphors [3] is based on an analogy with something in the physical world. Figure 1 is illustrating a conceptual model, a floating elastic ball in the water, for a fisheye lens. So in this analogy, changes in the height of the center of the ball outside the water, $y(t)$, adjusts the degree of magnification (DOM) and is function of time $\dot{y}(t)$. If we show the radius of the ball with R (maximum DOM), then

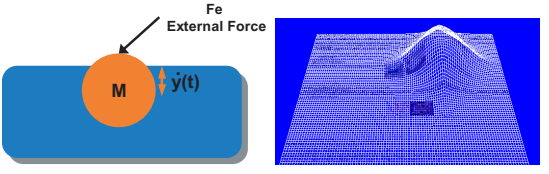
$$DOM(t) = R - \dot{y}(t) \tag{1}$$

When we apply an external force, f_e, we push the ball down in the water (not more than its radius) so the DOM decreases and when we release the force the DOM starts to increase (not more than its radius, see Figure 1). So the DOM is a variable which is continuously controlled by external force (mouse or tilting angles) and speed of movement. From Newton's second law of motion we can write the equation in vertical direction:

$$m\ddot{y}(t) = f_y - k\dot{y}(t) \tag{2}$$



(a) Left-A floating ball in the water. Right- A fully magnified fisheye lens.



(b) Left-The ball after applying external force and pushing it down in the water. Right- The DOM decreases as soon as the user starts to scroll.

Fig. 1. Interpreting Fisheye Lens as a floating ball

k is the damping factor caused by water resistance, and the effect of gravity and the weight of ball is negligible. In the horizontal direction we can write:

$$\begin{aligned} ma &= f_x - kv \qquad\qquad or \\ a &= \frac{f_x}{m} - \frac{k}{m}v, \end{aligned} \tag{3}$$

where v and a represent velocity and acceleration and k is the damping factor caused by water resistance. We may assume f_x is a function of f_y and velocity (this assumption will couple rates of change in DOM to speed of movement, as well as input) as below:

$$f_y = cf_x - bv \tag{4}$$

Where c and b are coefficients. After substituting f_y in (2) we can rewrite it as below:

$$\ddot{y}(t) = \frac{c}{m}f_x - \frac{b}{m}v - \frac{k}{m}\dot{y}(t) \tag{5}$$

From classical textbooks in control theory [13] we can represent the mathematical model of our physical system as a set of input, output and state variables related by first-order differential equations in a state-space model. If we introduce x as position then velocity and acceleration will be first and second derivatives of the position respectively. The chosen state variables are $x_1(t)$ as position of

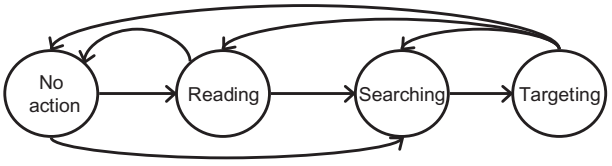


Fig. 2. Four discrete states of control mode in text-browsing example and transitions among them

cursor, $x_2(t)$ as velocity, $x_3(t)$ as rate of change of the DOM and u as f_x. So state variables can be written as below:

$$\dot{x_1}(t) = v = x_2(t) \tag{6}$$

$$\dot{x_2}(t) = a = \dot{v} = \frac{-k}{m}x_2(t) + \frac{u(t)}{m} \tag{7}$$

$$\dot{x_3}(t) = \ddot{y}(t) = \frac{-b}{m}x_2(t) + \frac{-k}{m}x_3(t) + \frac{c}{m}u(t) \tag{8}$$

The standard matrix format of these equations is:

$$\begin{pmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{-k}{m} & 0 \\ 0 & \frac{-b}{m} & \frac{k}{m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \\ \frac{c}{m} \end{pmatrix} u \tag{9}$$

This matrix reproduces the standard second-order dynamics of a mass-spring-damper system which we used previously [14]. Also this has many parameters that can be tuned, usually as a series of interacting, but essentially separate equations. Here, a 2 degree of freedom input can control both velocity and magnification factor so it proves a simple dynamic model can be tuned for different interactive models and generate different behaviors in controlling the task (next section). For example, the focus-targeting problem [15] can easily be solved in state-space representation by tuning c in matrix A or the 'hunting effect' problem [15] when the user overshoots the target due to the system increasing the DOM as the user slows, becomes a matter of tuning the dynamics of the system by changing the entries in the A matrix (For more information refer to [14]).

3 User Behavioral Models

In the 60's and 70's William Powers suggested [16,17] that many kinds of behavior can be described as control systems, and he argued that behavior is not output but, is the *control* of perception. In the model-based text browser example, the user's input, mouse data, controls what s/he perceives via focus-in-context and sonification feedback. In this example we assume the user is acting in one of four different modes: *no-action, reading, searching and targeting.*

Figure 3 illustrates the general framework. Figure 3(b) shows the classification of the user behaviour being used to switch the control mode. This mode is

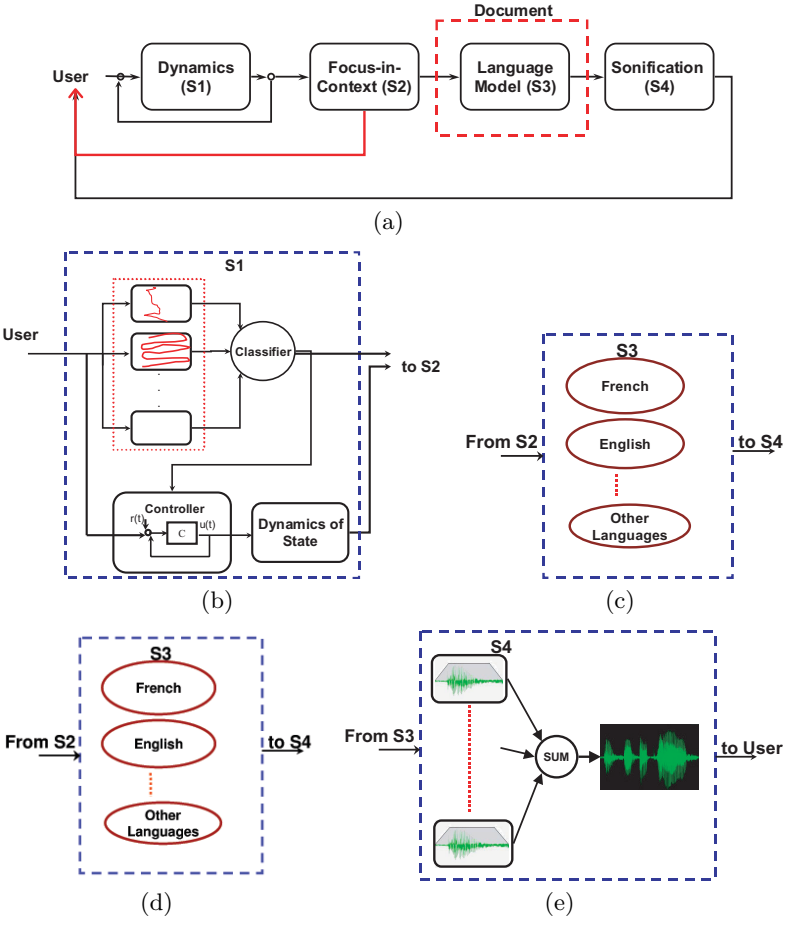


Fig. 3. (a) A general probabilistic framework of the model-based behavior system. (b) A Bayesian classifier classifies the user's input. Its output and the user's input come to the controller and change the dynamics (state variables). (c) State variables coupled to the focus-in-context change the size and shape of lens. (d),(e) The language identification method infers the most probable language inside the window around the lens and its output probabilities are fed to the audio synthesis algorithm.

then coupled to the visualization parameters, as shown in Figure 3(c), where the control mode changes the size and shape of the lens, and the controller provides the DOM, position and speed of the lens. For example, in reading mode the controller adjusts the DOM to stay in the maximum level, but as a long horizontal lens, while if the user 'breaks out' into general searching, the DOM is decreased smoothly to a lower level. This prevents the targeting problem [15] in focus-in-context techniques.

3.1 Detecting State Transitions

Figure 2 illustrates the possible state transitions. Initially, the user is in the no-action state. Depending on the input behaviour, the user can either go to the reading or the searching mode. A qualitative description of the automatic mode transitions is given below: In the reading mode, the user is making continuous increasing changes in x direction (left-to-right) and small changes in y direction (not more than height of a line) and at the end of the line makes a sudden change from right-to-left in x direction. If the changes in y direction are more than height of the line the system switches the mode to the searching mode.

After finding the target, the user slows down or stops scrolling until the lens is over the target point (targeting mode), or can return to the no-action mode directly.

A general technique for implementing this is to use a probabilistic classification of the likelihood of being in one of these four browsing behaviors according the joint probability of the input and output time-series. From Bayes' law, we write this as below:

$$P(Mode \mid X) = \frac{P(Mode)P(X \mid Mode)}{P(X)} \tag{10}$$

where X is an appropriate window of previous inputs and possibly also outputs. $P(X \mid Mode)$ can be identified from experimental data collected from test users using standard density estimation models.

3.2 Changing Meaning of Inputs

Given the inferred user task, the controller behaviour should be designed to support the user by enabling them to complete the task with as little effort as possible. This can include changing the interpretation of the inputs to being reference values, rather than direct control actions. Taking our inspiration from modern aircraft controllers, which have different interpretations of aircraft controls depending on flight mode (e.g. take off, altitude-hold, attitude-hold etc.), and which blend seamlessly between modes. See [18] for examples. For example, if the classifier infers that the user is in reading mode, then the controller automatically scrolls the lens from left to right and moves to the next line smoothly, rather than the user having to do this. Any left–right movement of the mouse now controls the reference reading speed that the reading mode controller is trying to achieve. Similarly other modes can reinterpret control inputs as browsing speed, or as position acquisition control while zooming in to a point of interest after browsing. This means that as the user performs the various tasks they switch between control modes automatically, and their inputs have different meanings, but that the transitions are always smooth and natural, and the user is often not even aware that their movements are having a different effect in the different modes.

4 Language Identification System

Language classification consists of two major stages. From Figure 4 we see at the top we have the modeling stage. During this stage, the language-specific features of a text are learned and stored in a model. First, as can be seen on the upper left-hand side in this figure, the distinctive features for each language in a multilingual corpus are determined and stored in a language model. Later, seen on the upper right-hand side, the features of a specific text are determined and stored in a document model. In this application a language model based on partial predictive matching [19] is used to calculate the probability of letter, *l*, through a conditional probability distribution *P(letter | prefix)*, which specifies the view about future possible value of *l*, conditional upon the truth of that particular description *prefix* on a per-word basis. Then a tree with probability information is generated from a corpus [20]. In our application these trees are built from short texts collected from *BBC* and *Le Monde* news web-sites in English and French (only few paragraphs). For simplicity no grammar or word-level model is used, although this would be likely to improve performance significantly [21]. At the bottom of the Figure 4, the classification stage is shown. During this stage, a word (the user is pointing to) of an untrained text in a document is compared to these trained language models. The language model which is the most similar to the language of this word is then selected, and represents the language of the word the user has pointed to. The actual comparison method depends on the classification technique used.

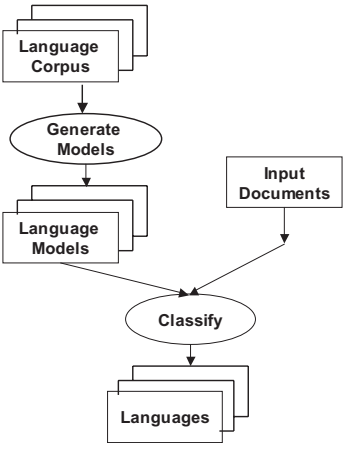


Fig. 4. The major stages of language identification system.(Top-left) The distinctive features for each language in a multilingual corpus are determined and stored in a language model tree. (Top-right) The word the user is pointing to in an untrained text is compared to the language models during the classification stage. The language model, which is the most similar to this word is then selected.

Language Prediction
Prediction in this application is done using Bayes' Law to infer the most probable language given text from a document.

$$P(Language \mid Word) = \frac{P(Language)P(Word \mid Language)}{P(Word)} \tag{11}$$

The document we have considered in this applications contains sentences and paragraphs both in English and French. When the user is scrolling over the text the application provides a virtual window (with the size of the lens' width, which is dynamic and adapts with any change to the DOM) around the cursor (Figure 3). Then the probabilistic language models calculate the probabilities of all words in the window in each language. For example, for only two words, $w1$ and $w2$ in the window, we have:

$$P(Language \mid w1, w2) = P(w1, w2 \mid Language).P(Language)/P(w1, w2) \tag{12}$$

As we have made the simplifying assumption that words in the window are independent, we can write the generalized form of equation (12) as below:

$$P(Language \mid Window) = \left[\prod_{i=1}^{i=n} \frac{P(w_i \mid Language)}{P(w_i)}\right] P(Language)$$
$$n \; is \; window \; size, \quad \forall \, i = 1 \; to \; n \quad w_i \in Window \tag{13}$$

So, we infer the language from a number of words from a document contained by the fisheye lens. The most probable language for any part of the text can be estimated as accurately as desired by making the window (or Drop-Off function's width in the fisheye lens [22]) sufficiently small.

5 Language Model and Granular Synthesis Feedback

As an intuitive model of the sonification process, we can imagine the words in the text to be embossed on the surface. Similar to [23] we simulate this model in our implementation by drawing an audio sample and placing that in an audio buffer, as each word belongs to a certain class of language "hits" the lens. This technique is a form of granular synthesis; [24] gives other examples of granular synthesis in interaction contexts. A real world analogy would be the perception of continuous levels of radiation via frequency of discrete pulses from a Geiger counter; here the continuous variable is the word flow rate in a specific language.

At a higher rate-of-scroll the acoustic response of the system, e.g. sampling frequency and volume of the audio sample decreases and provides the sense of distance to the text. At lower rates-of-scroll the sampling frequency and volume of the audio increases and the user feels he is getting closer to the text. Also, the volume and audio frequency are inversely related to the rate of scroll, so the audio texture as we pass over the text gives both an impression of the language of the text, as well as the speed at which we are passing it.

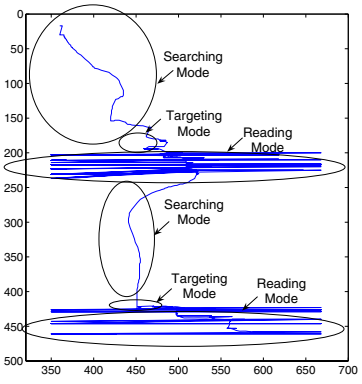
Similar to [24], the sonification technique can be extended to language recognition. We can sonify a probabilistic language recognizer by associating each language model with a source waveform, and each model's output probability then directly maps to the probability of drawing a grain from the source corresponding to that model (Figures 3(d) and 3(e)). The temporal distribution of grains inside the source waveforms maps to the probability of the language of the words inside the virtual window. The overall grain density is dynamic throughout the sonification when the user scrolls over the text. In practice, during the searching mode this produces a sound that's unclear when text features are blurred and the DOM is in the minimum level, and it means the information entropy inside the virtual window around the cursor is high. This features resolve to a clear, distinct sound as system's mode switches to the targeting. The sonification's primary effect is to display the current goal distribution's entropy, i.e. language, audio and text content.

The concept of entropy in information theory describes the level of uncertainty of a random variable. An alternative way to look at this is to talk about how much information is carried by the signal. For example, in an English text, encoded as a string of letters, spaces, and punctuation the signal is a string of characters. The letter frequency for different characters is different, and we cannot perfectly predict what the next character will be in the string: it is, to some degree, 'random'. Entropy is a measure of this randomness, suggested by Shannon [25].

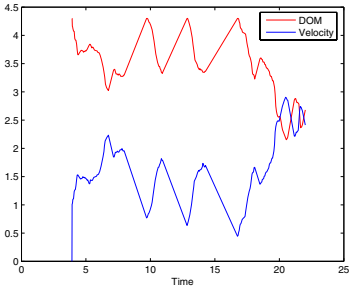
So model-based behavior in this task couples the user's input (speed of scroll) to the visualization technique via the dynamics and the focus-in-context method couples detail-in-context to audio samples via the language identification system (Figure 3(a)).

6 Example Use of Working System

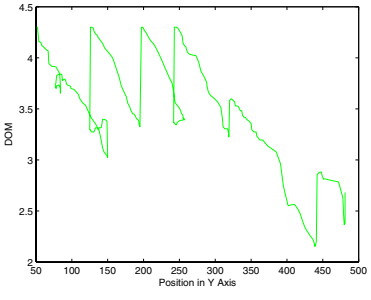
We developed a document viewer using the EPF library [22] for the focus-in-context method to browse a PDF, PS or DOC file which have been converted to an image (Bitmap) file. The document we presented was a 5 pages scientific document in English and a paragraph, a figure caption and few sentences written in French. The interaction is controlled via a mouse. The results in Figure 5(a) highlight the different navigation styles of the different interfaces and input methods. In the focus-in-context implementations the user had smooth navigation, which also included smooth changes in the DOM (See Figure 5(b)). If the velocity rises above a threshold DOM smoothly decreases and the reading mode switches automatically to searching mode, for instance in Figure 5(b) this has happened around t= 7, 11 and 14 seconds. So the velocity of the input device provides a smooth switch between different modes of control. Figure 5(c) presents how the DOM changes when the user has found the French sections in the document, stopped for a brief check and clicked over the text. The French sections are around pixels: 150, 190, 270, 330, and 420 and we see the user has found the most of sonically highlighted sections.



(a) The user's trace in the text browser example. The user starts the scrolling from top-left corner (beginning of the document) and scrolls down. The searching behavior becomes targeting and then reading behavior.



(b) Change in the DOM and velocity versus time.



(c) Change in the DOM versus position in finding French sections around pixels 120 ,190, 270, 330, and 420.

Fig. 5. Plot of logged data in searching French sections by one of users. Note that the presented document in (b) and (c) is different from the document in (a).

7 Conclusions and Future Work

In this paper we presented a novel approach to designing interaction between the user and the system. This approach is a model-based interactive method for browsing a multilingual text based on a language model, focus-in-context method and continuous interaction interface. We presented a floating ball model as an example of how the dynamic approach can be used creatively to design interaction, and suggest new metaphors. The state-space, dynamic system representation coupled the user's intention to the visualization technique via only the two degree of freedom mouse input, allowing the user to switch smoothly among reading, targeting and

searching modes by only moving the mouse. A probabilistic language model was used for online classification of the focus content in a multilingual input document.

Our probabilistic audio feedback based on granular synthesis is an example of a multimodal approach to sense different languages in the document. The focus-in-context method representing this document coupled details in context to audio samples via the language identification system. So the system could provide both visual and audio feedback to the user.

A motivating factor behind the approach in this paper is that we can in the long-term, potentially develop the dynamic systems simulation approach as a systematic approach to creating designs which can shape interaction and provide rich multimodal feedback, in the same way that has been successful in other areas of computing, where physics and model-based approaches revolutionized the field, such as ray tracing algorithms in computer graphics [26].

More refinement of the prototype system would be required, and a thorough usability study needed to determine the practical applicability of the specific interface described here, but some initial observations are made below. Initial informal evaluation of the implementation of sensing multilingual texts on a laptop instrumented with a mouse and headphone were positive, and users felt that this provided an intuitive solution to the problem of finding information in a particular language in a multilingual text without reading the text. Sonifying each language in the document gave users a sense of their motion through the document, which allowed them to continue their interaction while being involved in other tasks. The system allowed users to browse the document and locate targets (here the idea was searching and locating French written parts of the document) without looking at the screen. Supporting *intermittent interaction*, where a user can spend varying amounts of attention on interaction while carrying on with other activities, is very important for usable interaction, while on the move, making this approach interesting for use in mobile phones and small screen devices.

Acknowledgements

The authors gratefully acknowledge the support of SFI BRG project *Continuous Gestural Interaction with Mobile devices*, HEA project *Body Space*, and SFI grant 00/PI.1/C067, the IST Programme of the European Commission, under PASCAL Network of Excellence, IST 2002-506778. This publication only reflects the views of the authors.

References

1. McCullough, M.: Abstract Craft: Practical Digital Hand. The MIT Press (1998)
2. Kelley, C.R.: Manual and Automatic Control. John Wiley and Sons, Inc., New York (1968)
3. Beaudouin-Lafon, M.: Designing Interaction, not Interfaces. In: AVI '04: Proceedings of the working conference on Advanced visual interfaces. (2004) 15–22
4. Doherty, G., Massink, M.: Continuous Interaction and Human Control. In Alty, J., ed.: Proceedings of the XVIII European Annual Conference on Human Decision Making and Manual Control. (1999) 80–96

5. Faconti, G., Massink, M.: Continuous interaction with computers: Issues and Requirements. In C.Stephanidis, ed.: Proceedings of Universal Access in HCI. Volume 3., Lawrence Erlbaum Associates (2001)
6. Bederson, B.B.: Fisheye Menus. In: UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology. (2000) 217–225
7. Furnas, G.: Generalized Fisheye Views. In: Proceedings of CHI'86. (1986) 16–23
8. Lamping, J., Rao, R., Pirolli, P.: A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In: Proceedings of CHI 95. (1995) 401 – 408
9. Mackinlay, J.D., Robertson, G.G., Card, C.K.: The Perspective Wall: Detail and Context Smoothly Integrated. In: Proceedings of CHI'91. (1991) 173–179
10. Sarkar, M., Brown, M.H.: Graphical fisheye views of graphs. In Bauersfeld, P., Bennett, J., Lynch, G., eds.: Human Factors in Computing Systems, CHI'92 Conference Proceedings: Striking A Balance, ACM Press (1992) 83–91
11. Carpendale, M.S.T.: A Framework for Elastic Presentation Space. PhD thesis, Department of Computing Science, Simon Fraser University, Canada (1999)
12. Preece, J., Rogers, Y., Sharp, H.: Interaction Design: Beyond Human Computer Interaction. John Willey (2002)
13. Sheridan, T.B., Ferrell, W.R.: Man-Machine Systems: Information, Control, and Decision Models of Human Performance. MIT press (1974)
14. Eslambolchilar, P., R.Murray-Smith: Tilt-based Automatic Zooming and Scaling in mobile devices-a state-space implementation. In: Mobile Human-Computer Interaction MobileHCI 2004: 6th International Symposium. (2004) 120–131
15. Gutwin, C.: Improving focus targeting in interactive fisheye views. In: Proceeding of CHI'02. (2002) 267–274
16. Powers, W.T.: Living Control Systems: Selected papers of William T. Powers. The Control Systems Group Book (1989)
17. Powers, W.T.: Living Control Systems II: Selected papers of William T. Powers. The Control Systems Group Book (1992)
18. Tischler, M.B.: Advances in Aircraft flight Control. Taylor & Francis (1994)
19. Bell, T., Cleary, J., Witten, I.: Text Compression. Prentice Hall Advanced Reference Series. Prentice Hall (1990)
20. Williamson, J., Murray-Smith, R.: Dynamics and probabilistic text entry. In Murray-Smith, R., Shorten, R., eds.: Hamilton Summer School on Switching and Learning in Feedback systems. Volume 3355 of Lecture Notes in Computing Science., Springer-Verlag (2005) 333–342
21. Lesher, G., Rinkus, G.: Leveraging word prediction to improve character prediction in a scanning configuration. In: Proceedings of the RESNA 2002, Annual Conference. (2002)
22. Carpendale, S., Montagnese, C.: A framework for unifying presentation space. In: Proceedings of UIST'01. (2001) 82–92
23. Eslambochilar, P., Williamson, J., Murray-Smith, R.: Multimodal feedback for tilt controlled speed dependent automatic zooming. In: UIST'04: Proceedings of the 17th annual ACM symposium on User interface software and technology, (ACM)
24. Williamson, J., Murray-Smith, R.: Sonification of probabilistic feedback through granular synthesis. In: IEEE Multimedia. Volume 12, Issue 2. (2005) 45–52
25. C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal,* vol. 27, pp. 379-423 and 623-656, July and October, 1948. `http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`.
26. Foley, J., Dam, A.V., Feiner, S., Hughes, J.F.: Computer Graphics, reissued 2nd Ed. Addison Wesley, ISBN: 0201848406 (1995)

The NIST Meeting Room Corpus 2 Phase 1

Martial Michel[1,2], Jerome Ajot[2], and Jonathan Fiscus[2]

[1] Systems Plus, Inc., 1370 Piccard Drive - Suite 270, Rockville, MD 20850, USA
[2] NIST, 100 Bureau Dr - MS 8940, Gaithersburg, MD 20899, USA
{martial.michel,jerome.ajot,jon.fiscus}@nist.gov

Abstract. The Speech Group and Smart Spaces Lab of National Institute of Standards and Technology's Information Technology Laboratory have collaborated to collect a second phase of meetings in the NIST Meeting Data Collection Laboratory. The meeting laboratory, which was used to collect a 15 hour pilot corpus beginning in 2001, was updated with 7 HDV cameras and new head microphones for participants to collect a twenty hour corpus to support ARDA's Video Analysis and Content Exploitation (VACE) program. For the second phase of data collection, a strong emphasis was placed on high resolution video along with video calibration to enable various visual processing applications such as three dimensional person tracking. This paper documents the evolution of the meeting room to its current form and describes the key components of the phase II corpus.

1 Introduction

Huge efforts are being expended in mining information in newswire, news broadcasts, and conversational speech and in developing interfaces to metadata extracted in these domains. However, until recently, little has been done to address such applications in the more challenging and equally important meeting domain.

The development of smart meeting room core technologies that can automatically recognize and extract important information from multi-media sensor inputs will provide an invaluable resource for a variety of business, academic, and government applications. Beside core applications that form the first tier for corpus collection and management, such metadata will provide the basis for second-tier meeting applications that can automatically process, categorize, and index meetings. Third-tier applications will provide a context-aware collaborative interface between live meeting participants, remote participants, meeting archives and vast online resources. Given that the necessary core meeting recognition technologies are in a fledgling state, it is essential that these first tier technologies be developed before the higher tier applications can be made useful.

The meeting domain has several important properties not found in other domains and that are not currently being focused on in other research programs:

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 13–23, 2006.

- Multiple Forums and Vocabularies: Meeting forums range from very informal to highly structured. Likewise, meeting vocabularies vary widely depending on both the meeting topic and degree of shared context among the participants.
- Highly-Interactive/Simultaneous Speech: The speech found in informal meetings is spontaneous and highly interactive across multiple participants and contains frequent interruptions and overlapping speech. This poses great challenges to speech recognition technologies that are typically tailored for single speaker speech streams.
- Multiple Distant Microphones: Meetings are typically recorded with multiple distant microphones. Speech recognition systems generally work quite poorly with distant microphones. Moreover, techniques have yet to be developed which efficiently integrate input from multiple microphones and take advantage of their positioning to improve recognition quality.
- Multiple Camera Views: Meetings are often recorded with multiple cameras with different and sometimes overlapping views. Much like the multi-microphone challenge above, this permits/challenges the technology to integrate data from multiple video inputs to enhance the metadata that can be extracted from the meeting and improve recognition quality.
- Multi-Media Information Integration: It is impossible to develop a complete understanding of meetings without analyzing a number of different signal types simultaneously: audio, video and other information sources (devices/resources with which participants interact).

In this article we present the National Institute of Standards and Technology Meeting Room Project whose role is to facilitate such work. First we present the orignal room used for our Pilot Corpus, then we detail the more complex room that is used in our second Corpus.

2 Corpus 01's Meeting Room

2.1 About the Room

Starting in 2001, NIST constructed a Meeting Data Collection Laboratory to collect corpora to support meeting domain research, development, and evaluation. The lab is equipped to look and sound like a conventional meeting room. As such, sensors have been inconspicuously placed and noisy processors have been located outside of the room. The background noise level in the room has been measured at 42 dB A-weighted (with the video projector turned off).

The room is about 22 x 22 feet (6.7 X 6.7 meters) and can be configured for a variety of meeting forums (conference, round table, class room). However, to support the initial multi-microphone experiments, all of the meetings in the pilot corpus were collected using a single conference table configuration.

The Data Collection Lab contained a variety of microphones and several video cameras all collected using the NIST Smart Data Flow 1 architecture [1,2], was developed by the NIST Smart Spaces Laboratory. The architecture streamed and

captured all of the sensor data from all microphones and cameras on multiple separate data collection systems in a Smart Data format (*SMD*). This format adds a time stamp for each piece of data collected (one for each video frame, ...), making it possible to have data streams synchronized to within a few milliseconds of one another, and enabling a post resynchronization of multimedia sources to one another.

2.2 Sensor Setup

Placed inside the room were 5 cameras and 200 microphones (as can be seen in Fig. 1).

The cameras (Sony EVI-D30 motorized Pan/Tilt/Zoom NTSC analog video cameras) were mounted to the wall. Four of these cameras had stationary views of the conference table. The fifth camera was floating and used to follow a particular participant, view the whiteboard or the conference table itself, depending on the meeting forum. Data from the cameras were collected using an analog NTSC-based video frame grabber (Linux Media Labs 33) at 720x480 pixels and 29.97 frames per second, giving an MJPEG data composed of two interlaced JPEGs for each frame collected. Each frame was recorded separately as one entry in the NIST SMD file corresponding to the meeting and video camera being recorded.

The microphones in the room were made of two types of hardware:

1. One 24-channel 48-kHz/24-bit A/D (RME Hammerfall), on a single system (refered to as *cots*), used to record up to 23 channels :
   (a) Sixteen channels of wireless personal microphones so that participants were free to move about the room. The wireless system consisted of Shure UA845 antenna distribution system, Shure U1 body pack transmitter, and Shure U4D dual wireless receivers. Each of up to eight participants was equipped with two microphones, a noise-canceling Shure WCM-16 hypercardioid electret condenser headset, and a Shure WL-185 cardioid electret condenser lapel microphone attached to the wireless tansmitter packs.
   (b) 7 channels used by 4 different microphones placed on the conference table: 3 Audio Technica AT841a omni-directional condenser boundary mics (positioned at the center and ends of the table), and an Audio Technica AT854R 4-channel condenser boundary microphone (placed at the center of the table)[1]. All those microphones are fed into a Presonus M80 preamplifier with stereo summing bus.

   All 24 channels of the cots system were synchronized using an ADAT system (RME ADI-8 Pro 8-channel ADAT/TDIF AD/DA converter), and recorded into a NIST SMD format into one file corresponding to the meeting being recorded.
2. Three 64-channel (59 active) 22,050Hz/16-bit NIST Smart Spaces Lab Mark-II linear array microphones. These microphone arrays were added to support

[1] The AT854R is unusual in that it contains 4 separate cardioid boundary mics, together covering 360 degrees.

far field recognition experiments. Three prototype were positioned on pole mounts against the north, east and west walls of the room. Each array channel was collected by a distant A/D in a PC located under the raised floor. Unfortunately, these computers suffered from a variety of technical problems from the under-floor environment and we were unable to collect enough usable array data for distribution.

The Data Collection Facility also recorded the room's PC/projection screen using *Camtasia Studio*[3] (at 5 frames per second in a proprietary format). Unfortunately, this capability was added late in the pilot data collection cycle, so it was used in only a few of the pilot corpus meetings

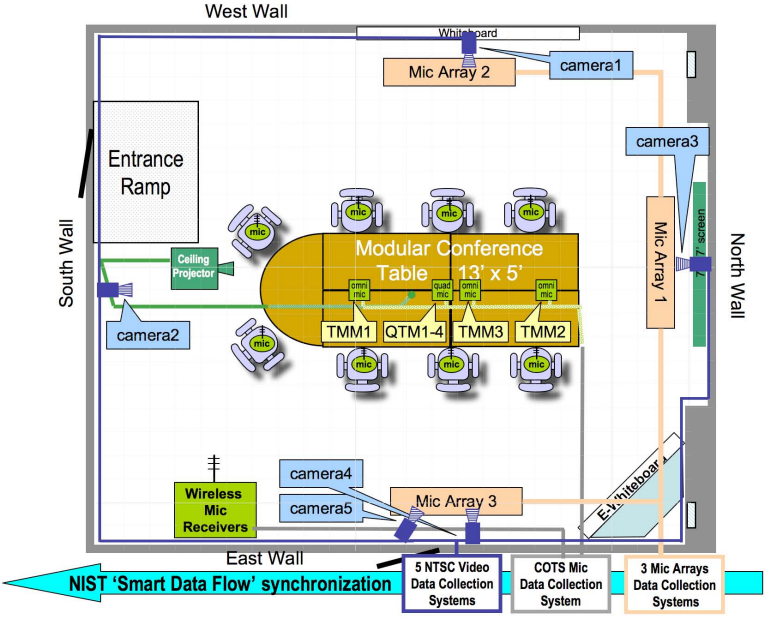


Fig. 1. Top view of Meeting Room 01

2.3 Data Distribution

The data were made ready for distribution so that all camera-captured video was converted to MPEG-2 for distribution. A synchronized audio channel was attached to the video; the left channel was a gain-normalized mix of all the recorded headset microphones and the right channel was a gain-normalized mix of all the recorded distant microphones. The Camtasia video screen capture recordings were also converted to MPEG2 for distribution. The individual head, lapel and table microphones were SPHERE-encoded, down-sampled to 16-kHz/16-bit and gain-normalized for distribution in single-channel files.

The NIST Meeting Room Pilot Corpus consists of 19 meetings, for a total of 15 hours per channel recorded between 2001 and 2003. In total, the multi-sensor data comes to 266 hours of audio [4] and 77 hours of video [5].

3 Corpus 02' s Meeting Room

3.1 Physical Environment Changes

The room itself is the same as one used in the original Corpus. The meetings in this Corpus have used some new layouts:

1. The conference configuration: same table layout as used for all meetings in Corpus 01 (the room and its new sensors can be seen in Fig. 2).
2. The classroom configuration: all the student tables are facing the west wall, where the white board is, and the teacher's table is next to the white board facing his students.
3. The discussion configuration: four tables are configured in a U shape, with the bottom and longuest part of the U toward the east wall, so that all participants have a view of the white board.

The room is now equipped with two different light systems (incandescent and fluorescent). Most of the meetings in this corpus have been recorded within the fluorescent light system environment, few of them with the incandescent light system, and some with both.

The ceiling projector (Proxima 6850+) has been replaced by a model (NEC GT6000) with both higher contrast and higher resolution. The A-weighted level of the room without the projector is 39.6dB, with the projector on, it becomes 43dB.

3.2 Sensor Changes

- The cameras in the room have been upgraded from MJPEG frame grabbed NTSC video frames to 7 Firewire captured MPEG2, using the JVC GR HD1 cameras. These cameras are digital HDV camcorders, recording 1280x720p MPEG2 TS, with GOP size of 6, at NTSC's 29.97fps, in a 16:9 aspect ratio, using 19.7 Mbps to transport the stream containing both an internal microphone and the video. Although these cameras provide us with far better video resolution and picture details, they are not Pan/Tilt/Zoom remote controlled. Therefore in order to have one camera focusing on a particular subject we had to use a positionner (Quickset QPT-15XD), and it is usually following the person seated at the corner of the southwest table. Camera 4, in Fig. 2, is looking down from the ceiling onto the southwest table, and is focused at the table level in order to give a proper view of that participant's hands.
- Microphone arrays are not present for this corpus. The SmartSpace's new generation Mk-III Microphone Array is available, but they are still being upgraded and will be integrated in the next corpus.

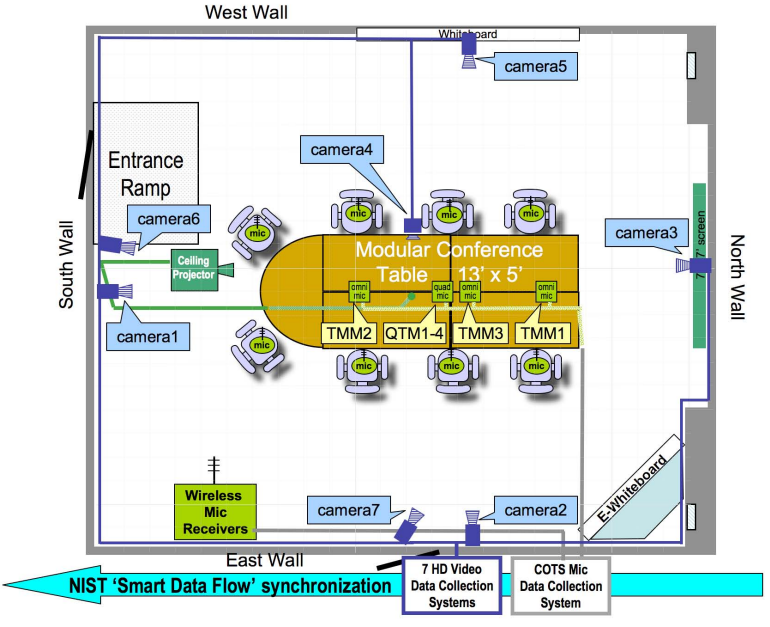


Fig. 2. Top view of Corpus 02's Meeting Room

– The cots sound recording equipment has stayed the same (RME Hamerfall captured ADAT synchronized wired and wireless microphones), except the head microphones which have been upgraded to Countryman Isomax E6 Directional EarSet Microphone for Shure Wireless (Model #E6DW5BSL). These head microphones are much smaller than the previous ones and do not obstruct the view of the participants' lips on the video. It is now possible to record up to 16 participants, all wearing head microphones (in Corpus 01, and when we record 8 or fewer participants, they wore both head and lapel microphones). To avoid participants encountering problems with the wireless packs (i.e. turning microphones off by mistake), the wireless battery packs have been placed in fanny packs.

3.3 Methodology Changes

Room and Meeting Artifacts. In this Corpus we have added a few room and corpus artifacts:

– A passport style photograph of each participant is taken, prior to recording.
– Posters, presentation boards and other text rich boards are placed on the walls, and replaced between meetings.
– Participants wear ID badge with fake names and different color schemes.
– As in Corpus 01, projector displayed computer screens are recorded using *Camtasia Studio*[3].

Camera Calibration. Camera calibration has been introduced in this corpus to integrate more information, accuracy and reliability to the video data. It provides information such as camera focus points, zoom values, and relative positions in the MEETING ROOM .

This corpus includes camera calibrations for every meeting using the camera calibration techniques developed by Bouguet [6]. The calibration has two main steps:

1. The intrinsic computation is the result of over 12 pictures taken by each camera with a planar checkerboard –of 14 rows and 20 columns, where each square measures 30x30mm– placed in full view on the camera frame. This computation provides each camera with its focal length, principal point, skew coefficient, and distortion coefficients.
2. The extrinsic calibration consists of taking one picture of another checkerboard –of 7 rows and 5 columns, where each square measures 90x90mm– in the same fixed position for all the cameras. Then using this picture and the results of the intrinsic calibrations for each camera, the relative position of the camera in the room is obtained. Information about the location of the board in the room is required to obtain accurate depth information.

Quality Control Checklists. Checklists have been created to ensure the quality of each recording. The checklists provide details on what needs to be done to successfully record a meeting, and includes :

- the day before the meeting: the full recording system is checked, the room is set for the type of meeting requested by the participants, cameras are checked anc calibrated, etc.
- the day of the meeting :
  - before the participants arrive: prepare the microphone packs, final room checks,etc.
  - before collecting, with participants in the room: take still photographs of participants, give them fake ID badges, provide them with instructions and details on the project,etc.
  - collect data.
  - after collection: name the meeting, compute synchronization values, compute intrinsic and extrinsic calibration,etc.
- after 5 business days have passed[2], the data are made available in a distribution-ready format.

Record and Review Station. The main challenge of recording so many sensors in parallel is being able to continuously check the status of all cameras and microphones.

[2] Human Subject Agreement gives participants 5 business days to review the data and inform us that they would like to have a portion of the meeting removed. Participants can also withdraw consent for the entire meeting.

Like the Pilot Corpus, we decided to use a software based solution relying on the SMART DATA FLOW to provide the recording operator with the means of seeing all 7 camera views as well as volume meters for all 24 individual microphones. Fig. 3 shows a screenshot of the "Review Station".

Each individual microphone volume meter has three color states to indicate the sound volume. The recording operator can listen to two individual channels being recorded (one on each ear), as well as control the output volume.

All 7 video views are presented at a reduced size; the real video size for such a frame is 1280x720, the reduced-size main camera is seen at 960x540 and the other 6 camera views are seen at 320x180. The user chooses the camera displayed on the main view. Since it is not possible to display all camera views at full speed[3], only I frames (5 per second) are displayed. The operator can select the main camera view to show in full resolution of 1280x720 at full frame speed (NTSC's 29.97fps).

In order to achieve this capture process, each individual sensor is captured on its own computer, interconnected with the SMART DATA FLOW (Figure 4 shows the capture map) :

- For each of the 7 cameras, a *Firewire MPEG2 RAW video capture* client provides a multiplexed audio and video MPEG2 TS flow to the *RAW MPEG2 TS Demuxer* which in turn splits the audio and video, writes frame-per-frame compressed MPEG2 ES video to SMD files, and transmit an *I Frame only* and *all Frames* flows.
- For the *RME capture client*, all 24 channels are captured multiplexed. This data is stored to disk in SMD files, and made available as data provided to both a *RME Volume* client (which creates the data used to draw the volume meter), and a *RME Stereo Extraction* client (whose role is to use the command feedback from the *Record Station* to extract two channels and provide them to the *Stereo Audio Play* client for audio playback of the selected channels).
- The *Review Station* receives most flows generated and displays them onto the screen on the command-and-control interface shown in Fig. 3.

Data are collected onto each individual capture system hard drive at 8 to 10 GB per hour per video camera and about 12 GB per hour for the RME audio capture client. We therefore record about 82GB of data per hour.

3.4 Corpus 02's Details

Preparation for Data Release. Once the meetings are captured to hard drives in the SMD format, they are first resynchronized[4], then trimmed to contain only the meeting itself. Data are then made distribution-ready:

[3] Having to transfer and decode 7 full speed 1280x720 cameras on an RGB surface would require extensive bandwidth and processing power.

[4] The full resynchronization process, and associated tools are described in another article[7], soon to be published. Check the SMART SPACE [15] and MEETING ROOM [16] websites for information.

Fig. 3. Record Station Screenshot

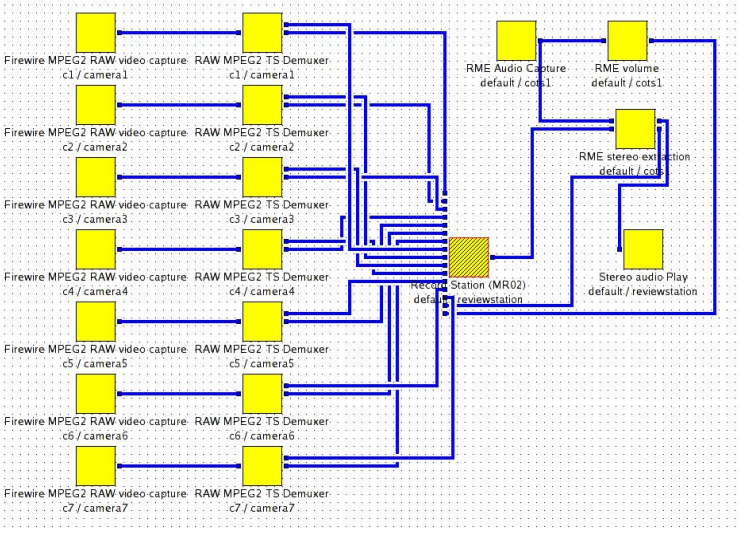


Fig. 4. Corpus 02 Capture Map

- For the video, this process involves multiplexing the trimmed MPEG2 video stream (result of the TS to ES demultiplex, at 1280x720, 16:9 aspect ratio, 29.97fps, 18.3Mbps) and the MPEG2 Audio Stream (Stereo, 48kHz, 256kbps) composed of an 80% gain normalized stereo mix of the head microphones on the left channel, and a volume normalized mix of the table microphones (not including the quad microphones) on the right channel.
- For the audio, each individual channel (head, lapel, omni, quad, ...) is converted from its 48kHz, 24 bit, linear PCM source format to 16kHz, 16 bit, linear PCM-sampled audio SPHERE-formatted files. A 50% gain normalization is applied to each channel.

Meeting Information. We took a different approach to selecting meetings to record for Corpus 02. We contracted with Enterprise Solutions Inc., a management consulting firm, and UserWorks Inc., a usability testing firm, to find groups willing to record naturally occurring, goal oriented meetings in our meeting room. This collaboration provided several meeting topics that theretofore had not been recorded as part of the meeting domain.

Meeting details will be posted on the Meeting Room website[16].

4 Conclusion

In this article, we have presented both the Meeting Room used in Corpus 01 and Corpus 02. We have detailed the differences in :

- Hardware: use of 720/30p HDV video cameras and head microphones that do not obstruct the participant's lips.
- Software: creation and evolution of capture clients and review station to insure proper use of new hardware.
- Methodology: with the creation of checklists insuring quality control during recording and processing of collected data, as well as camera calibration information.

Meetings details will be made available on the Meeting Room website[16].

In the future we intend to introduce participant localization information, use the finalized version of the Smart Space Mk-III mod 1 microphone arrays and go from 720p to 1080i HDV cameras.

References

1. M. Michel, V. Stanford, O. Galibert, (2003). Network Transfer of Control Data: An Application of the NIST Smart Data Flow. Proceedings of CCCT 2003.
2. M. Michel, V. Stanford, O. Galibert, (2003). Network Transfer of Control Data: An Application of the NIST Smart Data Flow. Journal of Systemics, Cybernetics and Informatics (Volume 2, Number 6).
3. Camtasia Studio. `http://www.techsmith.com/camtasia.asp`

4. J. Garofolo, M. Michel, V. Stanford, E. Tabassi, J. Fiscus, C. Laprun, N. Pratz, J. Lard (2004). NIST Meeting Pilot Corpus Speech. ISBN 1-58563-302-x. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004S09
5. J. Garofolo, M. Michel, V. Stanford, E. Tabassi, J. Fiscus, C. Laprun, N. Pratz, J. Lard, S. Strassel (2004). NIST Meeting Pilot Corpus Transcripts and Metadata. ISBN 1-58563-303-8 http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004T13
6. Jean-Yves Bouguet (N/A). Closed-form camera calibration in dual-space geometry. Technical notes, http://www.vision.caltech.edu/bouguetj/calib_doc/
7. M. Michel, V. Stanford (2006). Synchronizing Multimodal Data Streams Acquired Using Commodity Hardware. To be published.
8. J. Cugini, L. Damianos, L. Hirschman, R. Kozierok, J. Kurtz, S. Laskowski, J. Scholtz (1997). The Evaluation Working Group of the DARPA Intelligent Collaboration and Visualization Program.
9. J. E. McGrath (1984). Groups: Interaction and Performance. Englewood Cliffs, N. J., Prentice-Hall.
10. J. Garofolo, C. Laprun, M. Michel, V. Stanford, E. Tabassi (2004). The NIST Meeting Room Pilot Corpus. Proceedings of LREC 2004.
11. NIST (2002). Rich Transcription 2002 Meeting Recognition Evaluation, documentation. http://www.nist.gov/speech/tests/rt/rt2002/
12. NIST (2002). Rich Transcription 2002 STT and Metadata Extraction results, presentations, RT-02 Workshop. http://www.nist.gov/speech/tests/rt/rt2002/presentations/index.htm
13. NIST (2004) Rich Transcription 2004 Spring Meeting Recognition Evaluation, documentation. http://www.nist.gov/speech/tests/rt/rt2004/spring/
14. National Institute of Standards and Technology. http://www.nist.gov/
15. Smart Space. http://www.nist.gov/smartspace/
16. Meeting Room. http://www.nist.gov/speech/test_beds/mr_proj/
17. Systems Plus, Inc.. http://www.sysplus.com/

Disclaimer and License Statements. The Smart Data Flow software was developed at the National Institute of Standards and Technology by employees of the Federal Government in the course of their official duties. Pursuant to title 17 Section 105 of the United States Code this software is not subject to copyright protection and is in the public domain.

Certain commercial products may be identified in order to adequately specify or describe the subject matter of this work. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products identified are necessarily the best available for the purpose.

The Smart Data Flow is an experimental system. NIST assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

The National Institute of Standards and Technology and the Smart Space project would appreciate acknowledgements if the tools are used.

Audio-Visual Processing in Meetings: Seven Questions and Current AMI Answers

Marc Al-Hames[1], Thomas Hain[2], Jan Cernocky[3], Sascha Schreiber[1], Mannes Poel[4], Ronald Müller[1], Sebastien Marcel[5], David van Leeuwen[6], Jean-Marc Odobez[5], Sileye Ba[5], Herve Bourlard[5], Fabien Cardinaux[5], Daniel Gatica-Perez[5], Adam Janin[8], Petr Motlicek[3,5], Stephan Reiter[1], Steve Renals[7], Jeroen van Rest[6], Rutger Rienks[4], Gerhard Rigoll[1], Kevin Smith[5], Andrew Thean[6], and Pavel Zemcik[3,⋆]

[1] Institute for Human-Machine-Communication, Technische Universität München
[2] Department of Computer Science, University of Sheffield
[3] Faculty of Information Technology, Brno University of Technology
[4] Department of Computer Science, University of Twente
[5] IDIAP Research Institute and Ecole Polytechnique Federale de Lausanne (EPFL)
[6] Netherlands Organisation for Applied Scientific Research (TNO)
[7] Centre for Speech Technology Research, University of Edinburgh
[8] International Computer Science Institute, Berkeley, CA

Abstract. The project Augmented Multi-party Interaction (AMI) is concerned with the development of meeting browsers and remote meeting assistants for instrumented meeting rooms – and the required component technologies R&D themes: group dynamics, audio, visual, and multimodal processing, content abstraction, and human-computer interaction. The audio-visual processing workpackage within AMI addresses the automatic recognition from audio, video, and combined audio-video streams, that have been recorded during meetings. In this article we describe the progress that has been made in the first two years of the project. We show how the large problem of audio-visual processing in meetings can be split into seven questions, like "Who is acting during the meeting?". We then show which algorithms and methods have been developed and evaluated for the automatic answering of these questions.

1 Introduction

Large parts of our working days are consumed by meetings and conferences. Unfortunately a lot of them are neither efficient, nor especially successful. In a recent study [12] people were asked to select emotion terms that they thought would be frequently perceived in a meeting. The top answer – mentioned from more than two third of the participants – was "boring"; furthermore nearly one third mentioned "annoyed" as a frequently perceived emotion. This implies that many people feel meetings are nothing else, but flogging a dead horse.

⋆ This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811).

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 24–35, 2006.

Things get from bad to worse if transcriptions are required to recapitulate decisions or to share information with people who have not attended the meeting. There are different types of meeting transcriptions: they can either be written by a person involved in the meeting and are therefore often not exhaustive and usually from the particular perspective of this person. Sometimes they are only hand-written drafts that can not easily be shared. The second type are professional minutes, written by a person especially chosen to minute the meeting, usually not involved in the meeting. They require a lot of effort, but are usually detailed and can be shared (if somebody indeed takes the time to read over them). The third and most common transcript is no transcript at all.

Projects, like the ICSI meeting project [14], Computers in the human interaction loop (CHIL) [29], or Augmented Multi-party Interaction (AMI) [7] try to overcome these drawbacks of meetings, lectures, and conferences. They deal with the automatic transcription, analysis, and summarisation of multi-party interactions and aim to both improve the efficiency, as well as to allow a later recapitulation of the meeting content, e.g with a meeting browser [30]. The project AMI is especially concerned with the development of meeting browsers and remote meeting assistants for instrumented meeting rooms – and the required component technologies R&D themes: group dynamics, audio, visual, and multimodal processing, content abstraction, and human-computer interaction. "Smart meeting rooms" are equipped with audio-visual recording equipment and a huge range of data is captured during the meetings. A corpus of 100 hours of meetings is collected with a variety of microphones, video cameras, electronic pens, presentation slide and whiteboard capture devices. For technical reasons the meetings in the corpus are formed by a group of four persons.

The first step for the analysis of this data is the processing of the raw audio-visual stream. This involves various challenging tasks. In the AMI project we address the audio-visual recognition problems by formulating seven questions:

1. What has been said during the meeting?
2. What events and keywords occur in the meeting?
3. Who and where are the persons in the meeting?
4. Who in the meeting is acting or speaking?
5. How do people act in the meeting?
6. What are the participants' emotions in the meeting?
7. Where or what is the focus of attention in meetings?

The audio-visual processing workpackage within the AMI project aims to develop algorithms that can automatically answer each of these questions from the raw audio-visual streams. The answers can then be used either directly during or after the meeting (e.g. in a meeting browser), or as an input for a higher level analysis (e.g. summarisation). In this article we describe the progress that has been made in the first two AMI project years towards the automatic recognition from audio-visual streams, and thus towards answering the questions. Each of the next chapters discusses algorithms, methods, and evaluation standards for one of the seven questions and summarises the experiences we made.

2 What Has Been Said During the Meeting?

Meetings are an audio visual experience by nature, information is presented for example in the form of presentation slides, drawings on boards, and of course by verbal communication. The latter forms the backbone of most meetings. The automatic transcription of speech in meetings is of crucial importance for meeting analysis, content analysis, summarisation, and analysis of dialogue structure. Widespread work on automatic speech recognition (ASR) in meetings started with yearly performance evaluations held by the U.S. National Institute of Standards and Technology (NIST) [27]. This work was initially facilitated by the collection of the ICSI meeting corpus [14]. Additional meeting resources were made available from NIST, Interactive System Labs (ISL) [4] and the Linguistic Data Consortium (LDC), and more recently, the AMI project[7].

The objectives for work in ASR in meetings are to develop state-of-the-art speech recognition technology for meeting transcription; to enable research into meeting relevant topics into ASR; to provide a common working base for researchers; and to enable downstream processing by providing automatically annotated and transcribed data. All of these objectives require a common and standardised evaluation scheme and unified testing procedures. For ASR in general evaluations by word error rate measurement according to NIST protocols is standard. A more critical issue is the task definition with respect to input media and objective system output.

To ensure that the technologies under development are state of the art we participated in international evaluations of ASR systems for meeting transcription [27]. World-leading research groups in ASR enter this competition which aims to provide a comparison between different approaches by provision of standardised common data sets, an evaluation schedule, and by organisation of a workshop to ensure information exchange. AMI has successfully competed in the NIST RT05s STT evaluations [27], yielding very competitive results on both conference meeting and lecture room transcription [10, 11]. AMI specific evaluations are performed on AMI data alone. As all microphone conditions are available for the complete corpus no special targeted sub-sets are defined. In the course of our next development cycle we will implement a larger development test set (planning of this set had an input on data collection) that will cover all aspects of the corpus in terms of meeting room, scenario and speaker coverage.

Table 1 shows WER results for the 2005 AMI meeting transcription system. The high deletion rate is a main contributor to the error rate. The associated results on rt05seval MDM are also shown in Table 1, again with relatively high deletion rates. Particularly poor performance on VT data has a considerable impact on performance (only two distant microphones).

In summary, in the last two years we have defined an evaluation framework that is generic, flexible, comparable, and that allows us to conduct research and development in a stable environment. We have built a system that is very competitive and performs exceptionally well on AMI data. We can focus our attention on extending our work to the full AMI corpus and the specific problems to be faced there. Further a research infrastructure is in place that allows all

Table 1. WER in % on rt05seval IHM, respectively MDM

	TOT	Sub	Del	Ins	Fem	Male	AMI	ISL	ICSI	NIST	VT
rt05seval IHM	30.6	14.7	12.5	3.4	30.6	25.9	30.9	24.6	30.7	37.9	28.9
rt05seval MDM	42.0	25.5	13.0	3.5	42.0	42.0	35.1	37.1	38.4	41.5	51.1

partners to work on subtasks without the need to build large and labour-intensive systems or work on oversimplified configurations.

Our research results also give a better understanding of many interesting questions such as the distant microphone problem, the segmentation problem, the use of language, or the presence of many accents. Our investigations highlight where major improvements in performance can be obtained.

3 What Events and Keywords Occur in the Meeting?

Acoustic event and keyword spotting (KWS) are important techniques for fast access to information in meetings. Here we will concentrate on KWS: the goal is to find the keyword and its in speech data including its position and confidence. In AMI we compared three approaches to KWS. They are based on a comparison of two likelihoods: that of the keyword and the likelihood of a background model.

In the *acoustic approach*, phoneme-state posteriors are first estimated using a system based on neural networks with split temporal context [21]. The models of keywords are assembled from phoneme models and run against a background model (a simple phoneme loop). The difference of two log-likelihoods at the outputs of these models forms the score. It is advantageous to pre-generate the phoneme-state posteriors. The actual decoding is then very fast. We have further accelerated the decoding by pruning the phoneme-state posterior matrices by masking them using phoneme lattices discussed below. Then the decoding runs about $0.01 \times$ RT on a Pentium 4 machine. *KWS in LVCSR lattices* greps the keywords in lattices generated by a large vocabulary continuous speech recognition system (LVCSR, Sect. 2). The confidence of each keyword is the difference of the log-likelihood of the path on which the keyword lays and the log-likelihood of the optimal path. The *KWS in phoneme lattices* is a hybrid approach. First, phoneme lattices are generated. This is in fact equivalent to narrowing the acoustic search space. The phonetic form of the keyword is then grepped in such lattices and the confidence of keywords is given by the acoustic likelihoods of individual phonemes, again normalised by the optimal path in the lattice.

A detailed description of the different systems, features, and a comparison of neural networks and GMMs in acoustic KWS can be found in [25]. Table 2 presents results of the three approaches on three test-sets. The sets are carefully defined on the ICSI meeting database [14]. While "Test 17" contains 17 common words, the sets "Test 1 and 10" concentrate on rare words occurring at most

Table 2. Comparison of Figure-of-Merit (FOM) measure (in %) of KWS approaches

Test set	Acoustic	Word lattice	Phoneme lattice
Test 17	64.46	66.95	60.03
Test 10	72.49	66.37	64.1
Test 1	74.95	61.33	69.3

one, respectively ten times in the test set. The results confirmed our previous assumptions about the advantages and drawbacks of the different approaches:

LVCSR-KWS is fast (lattices can be efficiently indexed) and accurate, however only for common words. We see a clear degradation of performance for the sets "Test 1 and 10". We should take into account that less common words (such as technical terms and proper names) carry most of the information and are likely to be searched by the users. LVCSR-KWS has therefore to be completed by a method unconstrained by the recognition vocabulary. *Acoustic KWS* is relatively precise (the precision increases with the length of the keyword) and any word can be searched provided its phonetic form can be estimated. This approach is ideal for on-line KWS in remote meeting assistants, but even with the mentioned high speed of 0.01 $\times$ RT, it is not suitable for browsing *huge* archives, as it needs to process all the acoustic (or at least posterior probabilities) data. *Phoneme lattice KWS* is a reasonable compromise in terms of accuracy and speed. Currently, our work on indexing phoneme lattices using tri-phoneme sequences is advancing and preliminary results show a good accuracy/speed trade-off for rare words.

With the acoustic keyword spotter, an on-line demo system was implemented. This system uses a new closed-form based algorithm for speaker segmentation which takes into account time information of cross-correlation functions, values of its maxima, and energy differences as features to identify and segment speaker turns [16]. As for LVCSR spotting, it was completed by an indexation and search engine [8] and integrated into the AMI multimodal browser JFerret [30].

Future work includes improvement of the core algorithms and on KWS enhanced by semantic categories.

4 Who and Where Are the Persons in the Meeting?

To browse meetings and relate different meetings to each other it is important to know, who was actually in the meeting. In this section we first address the problem of identifying persons and then track them through the meeting. Once identified, we aim to track the persons location through the meeting room. The location of each meeting participant at each time instance is rather uninteresting for a later comprehension of a meeting. It is very unlikely that a user will browse a meeting and ask for the "three dimensional coordinates of participant A at time instance 03:24:12". However, while usually not used directly, the correct coordinates of each person in the meeting are an essential input to various other meeting analysis tasks, including the focus of attention (Sect. 8) and action

recognition (Sect. 6). Furthermore these methods rely on very exact coordinates; wrong coordinates will lead to an error propagation, or in the worst case, to a termination of subsequent tasks. Thus determining the correct location of each meeting participant at each time in the meeting is a very crucial task.

An identification of meeting participants is possible from both the face and the voice. Here we'll concentrate on the face. During recent international competitions on face authentication [15], it has been shown that the discriminant approaches perform very well on manually localised faces. Unfortunately, these methods are not robust to automatic face localisation (imprecision in translation, scale and rotation) and their performance degrades. On the opposite, generative approaches emerged as the most robust methods using automatic face localisation. This is our main motivation for developing generative algorithms [6, 5]. For AMI we proposed to train different generative models, such GMMs, 1D-HMMs, and P2D-HMMs, using MAP training instead of the traditionally used ML criterion. Currently, we are evaluating the algorithms on a face verification task using the well-known BANCA benchmark database [3]. Our results show that generative models are providing better results than discriminant models. The best results are achieved by P2D-HMM. However, it should be noted that P2-HMMs are also much slower than GMMs. The algorithms have been developed as a machine vision package for a well-known open source machine learning library called Torch vision [26]. This package provides basic image processing and feature extraction algorithms but also several modules for face recognition.

For localisation and tracking of the meeting participants we developed, applied, and evaluated four different methods. To evaluate these methods we used the AMI AV16.7 corpus. It consists of 16 meeting room sequences of 1-4 minutes length with up to four participants, recorded from two camera perspectives. The sequences contain many challenging phenomena for tracking methods, like person occlusion, cameras blocked by passing people, partial views of backs of heads, and large variations in the head size. A common evaluation scheme, based on the procedure defined in [23] and a defined training and test corpus, allows to compare the advantages and the drawbacks of the different methods.

The *trans-dimensional MCMC* tracker is based on a hybrid Dynamic Bayesian Network that simultaneously infers the number of people in the scene and their body and head locations in a joint state-space formulation [22]. The method performs best when tracking frontal heads in the far field view. The *Active Shape* tracker is based on a double layered particle filter framework, which on the one hand allocates sets of particles on different skin coloured blobs and evaluates predicted head-shoulder contours on the image data. Especially in scenes with partial occluded heads the tracking algorithm shows its great performance. The *Kanade-Lucas-Tomasi (KLT)* tracking uses an image pyramid in combination with Newton-Raphson style minimisation to find a most likely position of features in a new image [13]. This method tracks heads correctly in more than 90% of the video sequences, however hands are often misinterpreted as heads. The *face detector* is based on a skin colour blob extraction followed by a movement

prediction. The face detector is based on the weak classifier compound of a Gabor wavelet and a decision tree [19]. The negative aspect of this face detector is the strong computation dependency on the Gabor wavelet feature evaluation and therefore it can not be used in real-time applications.

A comparative study of the four different head tracking methods, a detailed descriptions of the algorithms, and evaluation results can be found in [24].

5 Who in the Meeting Is Acting or Speaking?

The objective of this work is to be able to segment, cluster and recognise the speakers in a meeting, based on their speech. Speaker information can be included in the meeting browser so that the user will have a better understanding of what is going on and will have a better context of the contents.

Within AMI we developed two approaches. The first uses the acoustic contents of the microphone signal to segment and cluster speakers. This extends earlier TNO work on speaker recognition (for telephone speech) and speaker segmentation/clustering (for broadcast news). The system has been evaluated in the NIST Evaluation on Meeting Data [27]. The evaluation set contained ten meetings in total, two meetings each from five different sources. One meeting source was AMI. We participated in both the Speech Activity Detection task and the Speaker diarisation task. The system obtained very competitive results in the NIST RT05s evaluation for speech activity detection (the lowest error rate reported) and our speaker diarisation system performed satisfactorily, given the technology we used.

The second system is a new closed-form localisation based algorithm which takes into account time information of cross-correlation functions, values of its maxima, and energy differences as features to identify and segment speaker turns [16]. In order to disambiguate timing differences between microphone channels caused by noise and reverberation, initial cross-correlation functions were time-smoothed and time-constrained. Finally we used majority voting based scoring approach to decide about the speaker turns.

The system was tested on challenging data recorded within the AMI project (ICSI, AMI-pilot, and BUT data) recorded at 16kHz. Achieved results show that the between-channel timing information brings sufficient information about speaker turns, especially in case of segmenting heavily cross-talked data. The achieved frame-level accuracy (for every 10ms) is around 90% for all three databases even thought the degree of the cross-talk (influencing the reliability of particular hypothesis) varies a lot between different meeting data.

The proposed system has been successfully applied to segment newly created real meeting-recordings for AMI. Obtained rough speaker-turns (with speech and silence segmentation based on classical MFCCs classified using neural network trained on ICSI training data set) are exploited by annotators to create word-level orthographic transcriptions of new AMI meeting data.

For demonstration purposes, we also developed a speaker segmentation system that is able to detect speaker turns in real time. The system has been proposed together with acoustic based key-word spotter (Sect. 3). Furthermore, on-line pre-processing of visual input from the camera, scanning the whole scene using the hyperbolic mirror has been used.

6 How Do People Act in the Meeting?

We aim to extract visual features from videos and develop methods to use them for the automatic recognition of important actions and gestures in meetings. We focus on semantic actions and gestures that indeed happen in meetings and that can be of potential use to the user of a meeting-browser or as a cue for higher-level tasks in group analysis. We have defined a set of actions and gestures that are relevant for meetings, these include hand, body, and head gestures. Examples are Pointing, writing, standing up, or nodding. Special attention has been paid to negative signals, i.e. a negative response to a yes-no question usually characterised by a head shake. This kind of gesture contains important information about the decision making in meetings, but can be very subtle and involve little head movement, making automatic detection very difficult.

For the *gesture segmentation* two methods were applied: Bayes Information Criterion and an Activity Measure approach. As features we used Posio [18] (cf. Sect. 8) to extract for each person in the meeting the 2D location of the head and hands, a set of nine 3D joint locations, and a set of ten joint angles. In addition we performed *classification of the segmented data*. Due to the temporal character of gestures we focused on different HMM methods.

The main conclusion regarding the automatic segmentation of gestures in real meetings is that it still a very challenging problem and the tested approaches do not give good segmentation performance for whole gestures, mainly due to the intrinsic structure of the gestures and the noise in the input features. An alternative approach is to develop segmentation algorithms for gesture parts and in preliminary evaluations this gave promising results.

Given this segmentation experience, the classification task was performed on manually segmented video streams. We found that a garbage model improves the recognition performance significantly. The HMM approaches gave a reasonable performance. Gestures like standing up (100% recognition rate) and the important speech supporting gestures (85%) reached results satisfactory for practical applications. However the results for the detection of negative signals were not significantly better than guessing. Detecting gestures such as shaking or nodding and negative signals is still a challenging problem that requires methods capable of detecting very subtle head movements.

In summary: important gestures and actions in meetings, such as negative signals are very hard to detect, as they can be very subtle. The standard algorithms used for artificial gestures – such as HMMs – can therefore not be applied directly to the meeting domain. Methods capable of detecting very small movements are required and have to be investigated in detail.

7 What Are the Participants Emotions' in Meetings?

Recent studies [12] on emotions in meetings showed that people are – of course – not showing all kind of emotions in meetings, but only a rather small subset like bored, interested, serious, etc. On the other hand some emotions, like sadness, are very unlikely to appear. Furthermore peoples' expression of emotions in meetings is rather subtle compared to artificial emotion databases (see [17] for a recent survey). The combination of these two fact makes the detection of emotions in meetings rather difficult and calls directly for special methods. Similar to our AMI experience with gestures and actions (Sect. 6) standard methods for emotion detection from acted databases can not be directly applied to meetings.

AMI therefore aims to develop special algorithms to estimate the meeting participants' emotion from the information of head- and body pose, gestures and facial expressions. Therefore, the development and enhancement of the corresponding algorithms is crucial for emotion recognition by visual input. A description of activities can be found in Sect. 6. Independently, works are going on to analyse facial expressions. Very recent investigations are based on an application of the AdaBoost [9] algorithm and its variants applied on two-dimensional Haar- and Gabor-Wavelet coefficients, for localisation of frontal faces and eyes [28], as well as for classification of facial expressions [17]. Furthermore, an approach based on Active Appearance Models is implemented and investigated in its application to head pose estimation and facial expression analysis.

Evaluation of these – especially to the meeting domain adapted algorithms – is currently ongoing, showing very promising results. Even though this method shows high requirements to the computational performance of the applied hardware, the expected results argue for this approach.

8 Where or What Is the Focus of Attention in Meetings?

There are two questions to answer when trying to understand what is going on during the meeting. However, in view of the difficulty to determine both the group focus of attention (FOA) and the general FOA of individual people (a person might have multiple FOA – listening to a speaker while taking notes –, ground truthing a mental state is difficult), we restricted our investigations to the visual FOA of people defined as the spatial locus defined by the person's gaze, which is indeed one of the primary cue for identify the attentional state of someone [20]. With this definition, research was conducted into two directions.

In the first direction, the objective is to identify the role played by the FOA in the dynamics of meetings. Answering such questions will be useful to understand the relationship between the FOA and other cues (such as speaker turns, cf. Sect. 5) as well as to more precisely identify the interactions between participants (e.g. by contributing to the recognition of the higher level dialog acts), which in turn could translate to better FOA recognition algorithms. The second direction is concerned with the recognition of the FOA. More precisely, given recorded meeting data streams, can we identify at each instant the FOA of the meeting participants? Both directions were investigated and are summarised in four tasks.

Perception of head orientation in a Virtual Environment: This task consists of assessing how accurately people perceive gaze directions. In a virtual environment an avatar was positioned at one side of the table. At the other side a number of balls where placed at eye height for the avatar. Persons where then asked to predict at which ball the avatar was looking at. As a first result we found that there is no significant difference for the location of the avatar. Furthermore no learning effect among the participants has been found. Decreasing the angle between the balls increases the judgement error. With an azimuth angle between two persons at one side of the table of 30 degree, as seen from a person at the other side, an discrimination is possible with an accuracy of 97.57%. This shows that head orientation can be used as a cue for the FOA.

Identifying speaker amongst meeting participants: In this task AMI investigates, whether observers use knowledge about differences in head orientation behaviour between speakers and listeners by asking them to identify the speaker in a four-person setting. In a thorough study on the role of FOA in meeting conversations, we showed through the use of a Virtual Environment display that people are indeed using the gaze and head pose of participants to assess who is speaking. This results demonstrate that humans apply knowledge about systematic differences in head orientation behaviour between speakers and listeners. This shows how important the FOA in meetings is.

Head pose and head tracking: (cf. Sect. 4) One first step towards determining a person's FOA consists of estimating its gaze direction. Then from the geometry of the room and the location of the participants, the FOA can normally be estimated. However, as estimating gaze is difficult (and requires very close-up views of people to assess the position of the pupil in the eye globe), AMI has developed, as an approximation, algorithms for tracking the head and estimate its pose. We formulate the coupled problems of head tracking and head pose estimation in a Bayesian filtering framework, which is then solved through sampling techniques. Details are given in [2, 1]. Results were evaluated on 8 minutes of meeting recordings involving a total of 8 people, and the ground truth was obtain from flock-of-birds (FOB) magnetic sensors. The results are quite good, with a majority of head pan (resp. tilt) angular errors smaller than 10 (resp. 18) degrees. As expected, we found a variation of results among individuals, depending on their resemblance with people in the appearance training set.

Recognition of the FOA: In this task, the emphasis is on the recognition of a finite set $\mathcal{F}$ of specific FOA loci. Unlike other works, the set of labels in our setting was not restricted to the other participants, but included also looking at the table (e.g. when writing), at a slide screen, and an unfocused label (when looking at any direction different than those of the other labels). One approach to the FOA recognition problem that we have followed consists of mapping the head pose orientations of individual people to FOA labels. This was done by modelling each FOA with a Gaussian and the unfocus class with a uniform distribution. Evaluation was conducted on 8 meetings of 8 minutes on average. Each meeting involved 4 people, and the FOA of two of them was annotated.

First, we conducted experiments by using the head-pose pointing vectors obtained from the ground truth FOB readings. We obtained a frame-based classification rate of 68% and 47%, depending on the person's position in the smart meeting room. These numbers are lower than those reported in other works, and are mainly due to the use of a more complex setting, more labels, and demonstrate the impact of FOA spatial configurations on the recognition, and the necessity of exploiting other features/modalities (e.g speaking status) in addition to the head pose to disambiguate FOA recognition. Furthermore we found that using the estimated head-pose instead of the ground truth were degrading the results not so strongly (about 9% decrease, thus much less than the differences w.r.t. position in the meeting room), which was encouraging given the difficulty of the task. We also found that there was a large variation of recognition amongst individuals, which directly calls for adaption approaches like Maximum A Posteriori techniques for the FOA recognition. These adaptation techniques, along with the use of multimodal observation, will be the topic of current research.

9 Conclusion

In this article we described how audio-visual processing in meeting scenarios can be addressed with seven questions. We showed, how the project AMI developed and applied machine learning techniques to answer each of the questions automatically. By addressing the different audio-visual tasks with simple questions we were able to streamline and coordinate the development process and enable an easy sharing of data and recogniser outputs among the involved partners. This led to common evaluation schemes on commonly defined AMI data sets for each of the tasks and allows us to compare different approaches in a simplified way. Finally it is worth to mention, that this has been achieved by more than 50 persons from eight institutes in seven countries in the EU and the US.

References

[1] S.O. Ba and J.M. Odobez. Evaluation of head pose tracking algorithm in indoor environments. In *Proceedings IEEE ICME*, 2005.
[2] S.O. Ba and J.M. Odobez. A rao-blackwellized mixed state particle filter for head pose tracking. In *Proceedings of the ACM-ICMI Workshop on MMMP*, 2005.
[3] BANCA. Benchmark database. `http://www.ee.surrey.ac.uk/banca`.
[4] S. Burger, V. MacLaren, and H. Yu. The ISL meeting corpus: The impact of meeting type on speech style. In *Proceedings ICSLP*, 2002.
[5] F. Cardinaux, C. Sanderson, and S. Bengio. Face verification using adapted generative models. In *Int. Conf. on Automatic Face and Gesture Recognition*, 2004.
[6] F. Cardinaux, C. Sanderson, and S. Marcel. Comparison of MLP and GMM classifiers for face verification on XM2VTS. In *Proc. IEEE AVBPA*, 2003.
[7] J. Carletta et al. The AMI meetings corpus. In *Proc. Symposium on Annotating and measuring Meeting Behavior*, 2005.
[8] M. Fapso, P. Schwarz, I. Szoke, P. Smrz, M. Schwarz, J. Cernocky, M. Karafiat, and L. Burget. Search engine for information retrieval from speech records. In *Proceedings Computer Treatment of Slavic and East European Languages*, 2005.

[9] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, 1996.
[10] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, I. McCowan, D. Moore, V. Wan, R. Ordelman, and S. Renals. The 2005 AMI system for the transcription of speech in meetings. In *Proc. of the NIST RT05s workshop*, 2005.
[11] T. Hain, J. Dines, G. Garau, M. Karafiat, D. Moore, V. Wan, R. Ordelman, and S. Renals. Transcription of conference room meetings: an investigation. In *Proceedings Interspeech*, 2005.
[12] D. Heylen, A. Nijholt, and D. Reidsma. Determining what people feel and think when interacting with humans and machines: Notes on corpus collection and annotation. In J. Kreiner and C. Putcha, editors, *Proceedings 1st California Conference on Recent Advances in Engineering Mechanics*, 2006.
[13] M. Hradis and R. Juranek. Real-time tracking of participants in meeting video. In *Proceedings CESCG*, 2006.
[14] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The ICSI meeting corpus. In *Proceedings IEEE ICASSP*, 2003.
[15] K. Messer, J. Kittler, M. Sadeghi, M. Hamouz, A. Kostyn, S. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, N. Poh, Y. Rodriguez, and J. Czyz et al. Face authentication test on the BANCA database. In *Proceedings ICPR*, 2004.
[16] P. Motlicek, L. Burget, and J. Cernocky. Non-parametric speaker turn segmentation of meeting data. In *Proceedings Eurospeech*, 2005.
[17] M. Pantic and L.J.M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE TPAMI*, 22(12):1424–1445, 2000.
[18] R. Poppe, D. Heylen, A. Nijholt, and M. Poel. Towards real-time body pose estimation for presenters in meeting environments. In *Proceedings WSCG*, 2005.
[19] I. Potucek, S. Sumec, and M. Spanel. Participant activity detection by hands and face movement tracking in the meeting room. In *Proceedings CGI*, 2004.
[20] R. Rienks, R. Poppe, and D. Heylen. Differences in head orientation for speakers and listeners: Experiments in a virtual environment. *Int. Journ. HCS*, to appear.
[21] P. Schwarz, P. Matějka, and J. Černocký. Hierarchical structures of neural networks for phoneme recognition. In *Accepted to IEEE ICASSP*, 2006.
[22] K. Smith, S. Ba, J. Odobez, and D. Gatica-Perez. Evaluating multi-object tracking. In *Workshop on Empirical Evaluation Methods in Computer Vision*, 2005.
[23] K. Smith, S. Ba, J.M. Odobez, and D. Gatica-Perez. Multi-person wander-visual-focus-of-attention tracking. Technical Report RR-05-80, IDIAP, 2005.
[24] K. Smith, S. Schreiber, V. Beran, I. Potúcek, and D. Gatica-Perez. A comparitive study of head tracking methods. In *MLMI*, 2006.
[25] I. Szöke, P. Schwarz, P. Matějka, L. Burget, M. Karafiát, M. Fapšo, and J. Černocký. Comparison of keyword spotting approaches for informal continuous speech. In *Proceedings Eurospeech*, 2005.
[26] Torch. `http://www.idiap.ch/~marcel/en/torch3/introduction.php`.
[27] NIST US. Spring 2004 (RT04S) and Spring 2005 (RT05S) Rich Transcription Meeting Recognition Evaluation Plan. Available at `http://www.nist.gov/`.
[28] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.
[29] A. Waibel, H. Steusloff, R. Stiefelhagen, and the CHIL Project Consortium. CHIL: Computers in the human interaction loop. In *Proceedings of the NIST ICASSP Meeting Recognition Workshop*, 2004.
[30] P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with Ferret. In *Proceedings MLMI*. Springer Verlag, 2004.

A Multimodal Analysis of Floor Control in Meetings

Lei Chen[1], Mary Harper[1], Amy Franklin[2], Travis R. Rose[3], Irene Kimbara[2], Zhongqiang Huang[1], and Francis Quek[3]

[1] School of Electrical Engineering, Purdue University, West Lafayette IN
chenl@ecn.purdue.edu, harper@ecn.purdue.edu
[2] Department of Psychology, University of Chicago, Chicago, IL
[3] CHCI, Department of Computer Science, Virginia Tech, Blacksburg, VA

Abstract. The participant in a human-to-human communication who controls the floor bears the burden of moving the communication process along. Change in control of the floor can happen through a number of mechanisms, including interruptions, delegation of the floor, and so on. This paper investigates floor control in multiparty meetings that are both audio and video taped; hence, we are able to analyze patterns not only of speech (e.g., discourse markers) but also of visual cues (e.g, eye gaze exchanges) that are commonly involved in floor control changes. Identifying who has control of the floor provides an important focus for information retrieval and summarization of meetings. Additionally, without understanding who has control of the floor, it is impossible to identify important events such as challenges for the floor. In this paper, we analyze multimodal cues related to floor control in two different meetings involving five participants each.

1 Introduction

Meetings, which play an important role in daily life, tend to be guided by a set of principles about who should talk when. Even when multiple participants are involved, it is fairly uncommon for two people in a meeting to speak at the same time. An underlying, auto-regulatory mechanism known as 'floor control' guides this tendency in human dialogs and meetings. Normally, only one participant is actively speaking; however, around floor control transitions, several participants may vie for the floor and so overlapped speech can occur. The active speaker *holds the floor*, and the participants all compete for and cooperate to share the floor so that a natural and coherent conversation can be achieved.

By increasing our understanding of floor control in meetings, there is a potential to impact two active research areas: human-like conversational agent design and automatic meeting analysis. To support natural conversation between embodied conversational agents and humans, it is important that those agents use human conversational principles related to the distribution of floor control so that they can speak with appropriate timing. Further, the same embodied cues (e.g., gesture, speech, and gaze) that are important for creating effective conversational agents (e.g., [4]) are important for understanding floor control and how

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 36–49, 2006.

it contributes to revealing the topical flow and interaction patterns that emerge during meetings. In this paper, we investigate multimodal aspects of floor control in meetings.

Historically, researchers in conversational analysis have proposed models to describe the distribution of floor control. One of the most commonly cited models was developed by Sacks et al. [21]. A basic principle of this model is that a conversation is built on *turn constructional units* (TCUs), which are typically complete units with respect to intonation contour, syntax, and semantics. A TCU may be a complete sentence, a phrase, or just a word. The completion of a TCU results in a *transition relevance place* (TRP), which raises the likelihood that another speaker can take over the floor and start speaking. It is held that hearers use various cues to predict the end of TCUs.

Most previous research related to floor control coordination has been on dialogs. A variety of multimodal cues involving syntax, prosody, gaze, and gesture have been investigated for turn-taking in dialogs [1, 6, 7, 14, 17, 25]. Syntactic completion and special phrases, like "you know," are useful syntactic cues for turn change [6]. Silent pauses, rises or falls of intonation, variation of speech rate, final lengthening, and other prosodic patterns are related to turn keeping or yielding [6, 7, 14, 25]. Gestures can also be used to yield the floor [6]. During floor transitions, it is common to observe short periods of mutual gaze between two adjacent turn holders followed by the next holder breaking this mutual gaze, a pattern called *mutual gaze break* [1]. This pattern occurs in around 42% of the turn exchanges [17].

Recently there has been increasing research interest in multiparty meetings. For example, simulation studies of group discussions have been carried out to investigate turn-taking models of meetings [18, 19]. To support research on automatic meeting analysis, several audio or multimodal meeting corpora have been collected, including the **ISL** audio corpus [3] from Interactive Systems Laboratory (ISL) of CMU, the **ICSI** audio corpus [16], the **NIST** audio-visual corpus [8], and the **AMI** audio-visual corpus [15]. With the availability of these data resources, researchers have begun to annotate them with various kinds of events, most commonly dialogue acts (DAs) [10, 22]. They have also begun to develop methods for detecting these events in meetings using speech and multimodal cues (e.g., [9, 15]).

1.1 Our Focus

Floor control is an important aspect of human-to-human conversation, and it is likely that multimodal cues involving gesture, speech, and gaze play an important role for predicting floor control structure. Although research on multimodal cues for floor control is still at an early stage, the increasing amounts of multimodal meeting data should stimulate future progress.

To better support work related to floor control in meetings, we have developed a nomenclature that is focused directly on floor control-related events. Although researchers have developed annotations that express the role of utterances in turn management, they do not completely cover the phenomena involved in

floor control management. Hence, in this paper, we define a new floor control annotation scheme that builds on the notion of a sentence unit, and then use the annotations to identify multimodal cues of importance for predicting floor control related events.

We describe the audio/video meeting corpus used in our investigations in Section 2.1 and the annotations that are used for analysis in Section 2.2. In Section 2.3, we raise some questions related to floor control in meetings and present our quantitative results. In Section 3, preliminary conclusions based on our analysis of the data are presented.

2 Meeting Analysis

2.1 Meeting Description

In this paper, we annotated and analyzed two meetings from the VACE multimodal meeting corpus [5]. This corpus was collected in a meeting room equipped with synchronized multichannel audio, video, and motion-tracking recording devices. In these recordings, participants (from 5 to 8 civilian, military, or mixed) engage in planning exercises. Using a series of audio and video processing techniques, we obtained word transcriptions and prosodic features, as well as $3D$ head, torso and hand tracking traces from video tracking and a Vicon motion capture system, all time synchronized.

The two meetings selected for the current study were named based on their recoding dates. The Jan07 meeting, recorded on January 7th, involves the exploitation of a foreign device. In this meeting, 5 military officers from several different departments (e.g., device testing, intelligence, engineering, user community (i.e., fighter pilot)) collaborated to plan the best way to evaluate the capability of the device and exploit it. Each participant represented the perspective of his/her department. The Mar18 meeting, recorded on March 18th, involves the selection of graduate fellowship recipients. In this meeting, 5 faculty members from Air Force Institute of Technology (AFIT) developed criteria for selecting 5 awardees from a pool of 15 applicants and then made the selections based on those criteria. Each participant, after reviewing the qualifications of 3 applicants, gave their opinions about the ranking of their candidates and their suitability for selection. After an initial round of presentations, the participants developed selection criteria and ranked all of the candidates accordingly. The Jan07 and Mar18 meetings differ. The Mar18 participants needed to consult application materials during their interactions and also made use of a white board to organize the selection process. Because these artifacts played an important role in this meeting, there was much less eye contact among participants than in the Jan07 meeting.

2.2 Data Preparation and Annotation

The data annotation procedures used in this investigation are depicted in Figure 1. Details related to each step are provided below.

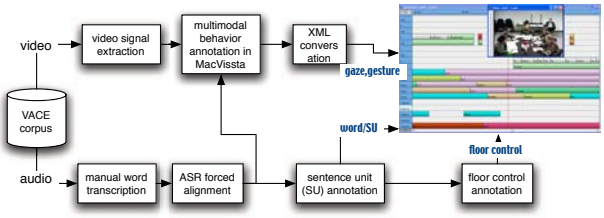


Fig. 1. Data flow diagram of our multimodal meeting data annotation procedures

Word/SU Annotation: The meetings in the VACE corpus were human transcribed according to the LDC Quick Transcription (QTR) guidelines and then time aligned with the audio. Word-level transcriptions do not provide information that is available in textual sources, such as punctuation or paragraphs. Because sentence-level segments provide an important level of granularity for analysis, we chose to segment the words into sentences and mark the type of sentence prior to carrying out floor annotation. The importance of structural information for human comprehension of dialogs has already been demonstrated and methods have been developed to automatically annotate speech dialogs [13]. Using EARS Metadata Extraction (MDE) annotation specification V6.2, we annotated sentence units (SUs). An SU expresses a speaker's complete thought or idea. There are four types: *statement*, *question*, *backchannel*, and *incomplete*.

The original EARS MDE SU annotations were created by LDC using a tool that was developed to annotate spoken dialogs. As we began this effort, there was no existing tool optimized to support the five-plus channels that must be consulted in our meetings in order to accurately annotate SUs. Furthermore, because we believed that the video cues were vital for our markups, we also needed access to the video. Hence, to annotate SUs (and subsequently floor control), we considered a variety of multimodal tools. We ended up choosing Anvil [12] for the following reasons: (1) it is an extremely reconfigurable tool developed to support multimodal annotations; (2) it supports the simultaneous display of annotations and playback of audio and video segments; (3) because it uses XML to represent the markups, the tool is able to flexibly support a variety of markups; (4) markups can be setup for color display, which is attractive especially for quickly post-editing annotations. For SU annotation, we used four different colors to differentiate among the SU types, which was quite helpful for noticing and correcting annotation mistakes.

Initially, we automatically annotated SU boundaries based on a hidden-event SU language model (LM) trained using the EARS MDE RT04S training data containing about 480,000 word tokens, and then we displayed this information using the Anvil interface designed by the first author. This interface displayed time aligned word transcriptions with the automatic markups and allowed the annotator to listen to audio and view video corresponding to selected portions of the transcripts in order to create gold SU annotations. Using this interface, the second author manually corrected SU boundary errors and added SU type

to each SU for both meetings. While carrying out these annotations, she also noticed and repaired a small number of transcription and word alignment errors.

Floor Control Annotation: There is no existing standard annotation for floor control, although LDC discussed the notion of a turn versus control of the floor in their annotation guidelines for MDE[1]. In previous research, most researchers have focused on turn-taking in dyadic conversations [2, 23, 24], but do not explicitly discuss the relationship between turn and floor control.

Following LDC's definition, we chose to define a speaker turn as an interval of speech uttered by a single discourse participant that is bounded by his/her silence (>= 0.5s). In contrast, the person controlling the floor bears the burden of moving the discourse along. When participant A is talking to participant B and B is listening without attempting to break in, then A clearly has "control of the floor"; however, other cases are less clear cut. Although turns are important for meeting analysis, not all speaker turns involve floor control, and it is possible to control the floor despite the presence of fairly long pauses. A participant's turn may or may not coincide with them holding the floor, and so may overlap with that of another participant who is holding the floor. Overlaps that do not cause the floor holder to concede the floor include backchannels (passive contributions to discourse, which constitute a speaker turn), failed interruptions, helpful interjections, or side-bar interactions. Change in control of the floor can happen through a number of mechanisms, including regular turn-taking and successful interruptions. We have developed an annotation scheme related to control of the floor that involves several types of events.

Control: This corresponds to the main communication stream in meetings. Since it is possible for floors to split, in our control annotations we keep track of who is in control and which participants are involved in each floor control event.

Sidebar: This event type is used to represent sub-floors that have split off of a more encompassing floor. Again we need to know who has control and which participants are involved.

Backchannel: This is an SU type involving utterances like "yeah" that are spoken when another participant controls the floor.

Challenge: This is an attempt to grab the floor. For example, the first utterance of "*do I*" by E is a challenge.

```
C: yeah we need to instrument it we need /-
E:                 do I...         do I need to be concerned...
```

Cooperative: This is typically a short utterance that is inserted into the middle of the floor controller's utterance in a way that is much like a backchannel but with helpful propositional content.

Other: These are other types of vocalizations, e.g., self talk, that do not contribute to any current floor control thread.

Using the Anvil interface designed by the first author which displays audio, video, and time aligned word transcriptions with SU annotations, the first author

[1] See https://secure.ldc.upenn.edu/intranet/Annotation/MDE/guidelines/2004/index.shtml

annotated each meeting segment with the above floor control related events, and these annotations were double checked by the second author. When annotating each event type we chose to respect the SU boundaries and event types. For example, SU backchannels must also be annotated as backchannels in our floor markups. Also control events respect SU boundaries (i.e., a control event continues until the end of an SU, even if another speaker starts a new control event before it is ended). The annotation process for these control events involved several passes. In the first pass, the annotator focused on tracking the "major" control thread(s) in the meeting, resulting in a sequence of floors controlled by various participants. Then, in the second pass, the annotator focused on the "finer" distinctions of floor control structure (e.g., challenge, cooperative). Anvil provides excellent play-back control to take some of the tedium out of viewing the data multiple times.

Gaze and Gesture Annotation: In each VACE meeting, 10 cameras were used to record the meeting participants from different viewing angles, thus making it possible to annotate each participant's gaze direction and gestures. Gesture and gaze coding was done on MacVissta [20], a general-purpose Mac OS X multimodal video display and annotation tool. It supports the simultaneous display of multiple videos (representing different camera angles) and enables the annotator to select an appropriate view from any of 10 videos to produce more accurate gaze/gesture coding. The annotators had access to time aligned word transcriptions and all of the videos when producing gaze and gesture annotations.

Following McNeill lab's gesture coding guidelines, five common types of gestures that are related to the content of cotemporaneous speech, including *metaphoric*, *iconic*, *emblematic*, *deictic* and *beat*, were annotated. These exclude fidgeting movements (e.g., tapping fingers while thinking, touching clothes) as well as instrumental movements (e.g., holding a cup, arranging papers on a desk). Gaze coding was completed by marking major saccades, which are intervals that occur between fixations of the eye. Such intervals begin with the shift away from one fixation point and continue until the next fixation is held for roughly 1/10th of a second (3 frames). Inclusion of micro-saccades is not possible using the available technologies nor is it necessary for our level of analysis. The segmentation of space into areas and objects for fixation include other people, specific non-human entities in the environment (e.g. board, papers, thermos), personal objects (e.g. watch), and neutral space in which the eyes are not fixated on any visible objects.

This gesture and gaze coding, which was stored using the Mac's default XML structure, was converted to a custom XML format that was then loaded into Anvil for combination with the word, SU, and floor control annotations described previously. Given the combination of word-level information, SU and floor event annotations, and gesture and gaze markups, we have carried out an analysis of the two meetings described previously.

2.3 Measurement Studies

We have two reasons for carrying out measurement studies on the two VACE meetings described above. First, since there has been little research conducted

Table 1. Summary statistics for two VACE meetings

Jan07 meeting							
speaker	dur(sec)	# words	Control	Challenge	Backchannel	Sidebar-Control	Cooperative
C	337.32	1,145	299.58 (37)	5.33 (8)	12.84 (44)	14.9 (17)	4.67 (4)
D	539.13	2,027	465.54 (26)	3.4 (7)	5.31 (29)	64.88 (19)	0 (0)
E	820.51	3,145	763.31 (63)	7.67 (11)	29.82 (116)	17.02 (7)	2.61 (2)
F	579.42	2,095	523.16 (37)	4.73 (20)	11.8 (43)	32.39 (15)	7.34 (9)
G	352.92	1,459	296.31 (31)	5.66 (11)	11.78 (55)	39.16 (15)	0 (0)
Mar18 meeting							
speaker	dur(sec)	# words	Control	Challenge	Backchannel	Sidebar-Control	Cooperative
C	679.39	2,095	648.73 (62)	1.89 (4)	28.02 (74)	0 (0)	0.75 (2)
D	390.46	1,285	359.75 (54)	4.23 (11)	21.78 (65)	0 (0)	4.7 (7)
E	485.21	1,380	465.03 (49)	10.24 (21)	18.41 (72)	0 (0)	0 (0)
F	486.60	1,467	481.70 (57)	1.43 (4)	3.47 (11)	0 (0)	0 (0)
G	470.72	1,320	422.49 (53)	0.87 (2)	36.76 (111)	0 (0)	2.14 (2)

on floor control in multiparty meetings, we need to gain a greater understanding of floor control in this setting. It is not clear whether the findings from dialogs will hold for larger groups of participants. Second, our ultimate goal is to develop algorithms that utilize multimodal cues to automatically annotate the floor control structure of a meeting. Hence, measurement studies provide an opportunity to identify useful cues from the audio and visual domains to support our future system design. Some of the questions that we had hoped to answer in this investigation include questions related to speech, gaze, and gesture. How frequently do verbal backchannels occur in meetings? What is the distribution of discourse markers (e.g., *right*, *so*, *well*) in the meeting data? How are they used in the beginning, middle, and end of a control event? When a holder finishes his/her turn, are there some observable distributional patterns in his/her eye gaze targets? Does he/she gaze at the next floor holder more often than at other potential targets? When a holder takes control of the floor, are there some observable distributional patterns in his/her eye gaze targets? Does he/she gaze at the previous floor holder more often than at other potential targets? Do we observe the frequent mutual gaze breaks between two adjacent floor holders during floor change? How frequently does the previous floor holder make floor yielding gestures such as pointing to the next floor holder? How frequently does the next floor holder make floor grabbing gestures to gain control of the floor?

Control of the floor is not always intentionally transferred from one speaker to another. Sometimes the floor holder yields the control of the floor and leaves it open to all meeting participants. When the floor is open, any of the meeting participants may take control without explicit transfer of control. In order to develop a better understanding of how gesture, speech, and gaze cues signal floor transitions, the first author classified all floor transitions into four categories:

Change: there is a clear floor transition between two adjacent floor holders with some gap between adjacent floors.

Overlap: there is a clear floor transition between two adjacent floor holders, but the next holder begins talking before the previous holder stops speaking.
Stop: the previous floor holder clearly gives up the floor, and there is no intended next holder so the floor is open to all participants.
Self-select: without being explicitly yielded the floor by the previous holder, a participant takes control of the floor.

For *Change* and *Overlap* floor transitions, the control of the floor is explicitly transferred from the previous to the next floor holder; whereas, for *Stop* and *Self-Select*, there is no explicit floor transition between the two adjacent floor holders. By distinguishing these four transition types, we believe we will be able to obtain a deeper understanding of the multimodal behavior patterns.

Basic Statistics: First, we provide some basic statistics related to floor control in the two meetings. Table 1 shows information about the Jan07 and Mar18 meetings. The table reports the total duration and the number (in parentheses) of each floor event type. It should be noted that these durations were calculated over intervals containing pauses. These meetings are clearly quite different based on the information provided in the table, even though each is comprised of five participants for a total duration of around forty minutes each. Figure 2 provides some basic statistics on floor transitions by meeting. We find that there is a much larger number of *Stop* and *Self-Select* floor transitions in the Mar18 meeting than in the Jan07 meeting.

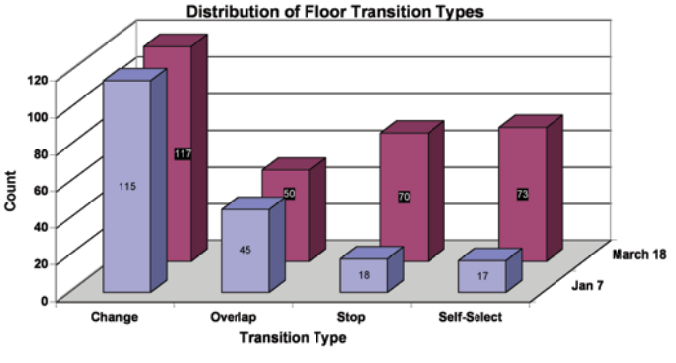


Fig. 2. Basic statistics on floor transitions for two VACE meetings

Speech Events: The first speech event we consider is the verbal backchannel. Given the fact that participants can make non-verbal backchannels (e.g., head nods) freely in meetings, we are interested in seeing whether verbal backchannels are common. Hence, we calculated the percentage SUs that are backchannel SUs. Figure 3 shows that the backchannel percentage is 25.22% in the Jan07 meeting and 30.8% in the Mar18 meeting. Jurafsky et al. [11] reported a backchannel percentage of 19% on the Switchboard corpus, and Shriberg et al. [22] obtained

a backchannel percentage of 13% on the ICSI meeting corpus. These percentages were calculated over utterances, where an utterance is a segment of speech uttered by a single speaker that is prosodically and/or syntactically significant within the conversational context [22]. In general, an SU may contain one or more utterance. Since nods made for affirmation purposes in the meeting were annotated, we include them here for comparison.

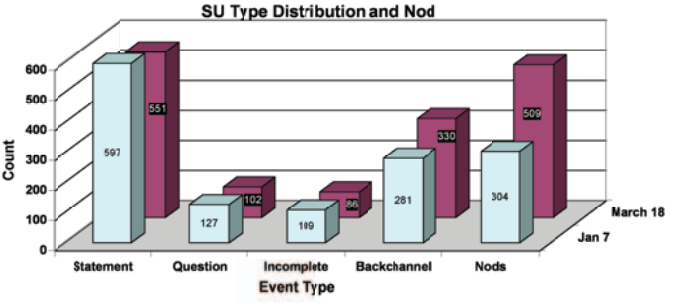


Fig. 3. SU type frequencies for two VACE meetings together with nodding frequency

Another important speech related feature that deserves some consideration is the patterning of discourse markers (DMs) in our floor control events. A DM is a word or phrase that functions as a structuring unit of spoken language. They are often used to signal the intention to mark a boundary in discourse (e.g., to start a new topic). Some examples of DMs include: *actually, now, anyway, see, basically, so, I mean, well, let's see, you know, like, you see* [13].

For *control* and *challenge* events, we counted the number of times that the DMs in the above list occur. For *control* events with durations exceeding 2.0

Table 2. DM distribution for Control and Challenge events in two VACE meetings

Jan07 meeting				
location	# w/ DM	#total	# dur. (sec)	frequency (Hz)
challenge	22	57	26.79	0.82
short control	20	54	52.41	0.38
beginning	58	140	70	0.82
ending	13	140	70	0.18
middle	304	140	2155.50	0.14
Mar18 meeting				
location	# w/ DM	#total	# dur. (sec)	frequency (Hz)
challenge	12	42	18.65	0.64
short control	42	110	111.04	0.38
beginning	73	165	82.5	0.88
ending	13	165	82.5	0.16
middle	184	165	2092.67	0.09

seconds, we count the number of discourse markers appearing in three locations, i.e., the beginning (the first 0.5 seconds of the span), the end (the last 0.5 seconds of the span), and the middle (the remainder). If a span is shorter than 2.0 seconds, we count the number of discourse markers appearing over the entire span and dub the event a *short control* event. Since floor *challenges* tend to be short, we also count the number of discourse markers over the entire span. Table 2 shows the distribution of DMs for these events and locations. We calculated the frequency of DMs, which is defined as the ratio of number of intervals with DMs (# w/ DM) to the total duration for a location or event (# dur. (sec)). DMs occur much more frequently in *challenges* (0.82 Hz in Jan07 and 0.64 Hz in Mar18) and in *floor beginnings* (0.82 Hz in Jan07 and 0.88 Hz in Mar18) than in the other event spans.

Gaze Events: Figures 4 and 5 report some statistics related to gaze targets of the previous and the next floor holders during a floor transition. The possible targets include the next holder, previous holder, the meeting manager (E in each meeting), other participants, no person (e.g., papers, object, whiteboard). In the Jan07 meeting, when the floor is transferred, the previous floor holder frequently gazes to the next floor holder (in 124 out of 160 transitions, giving 77.5%). In addition, the next floor holder frequently gazes at the previous floor holder (136 out of 160 transitions or 85%) during a floor transition. In the Mar18 meeting, since participants often spend time reading information from papers and the whiteboard, we find a much lower occurrence of these gaze patterns; the previous holder gazes to the next holder 65 times out of 167 transitions (38.9%) and the next holder gazes to the previous holder 76 times out of 167 transitions (45.5%).

In the Jan07 meeting, over all of the 160 floor transitions involving two holders (*Change* and *Overlap*), there were 70 mutual gaze breaks, giving a percentage

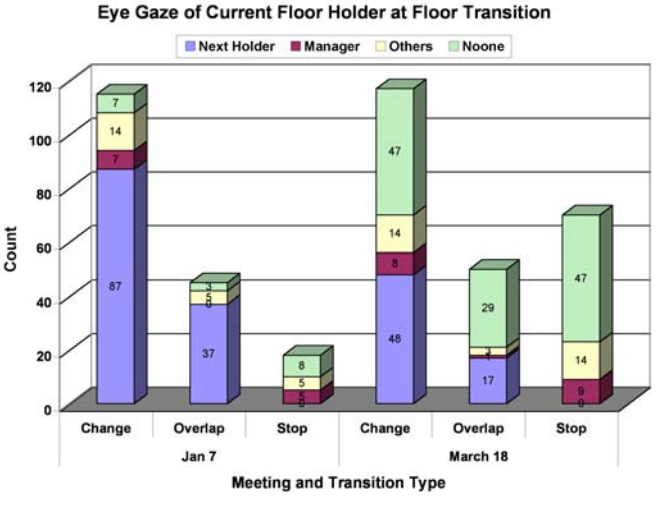


Fig. 4. The previous floor holder's gaze target distribution

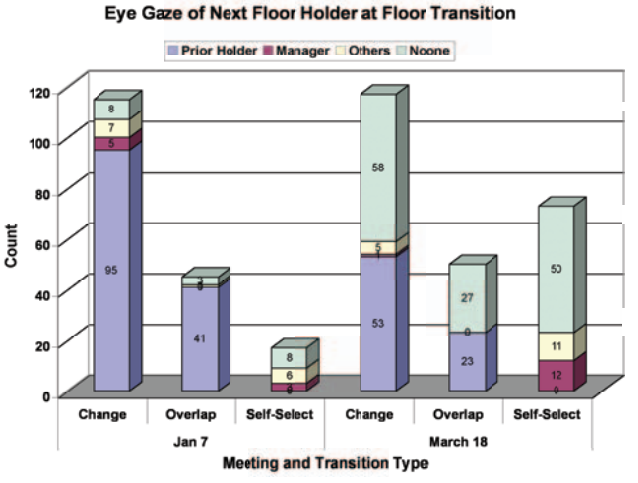


Fig. 5. The next floor holder's gaze target distribution

of 43.75%, which is similar to the 42% reported by Novick [17]. However, in the Mar18 meeting, over all 167 floor exchanges involving two holders (*Change* and *Overlap*), there were only 14 mutual gaze breaks. This suggests that in meetings that involve significant interactions with papers and other types of visual displays, there is likely to be a lower percentage of mutual gaze breaks.

All participants do not play equal roles in the meetings we analyzed; there was a meeting manager assigned for each meeting. The participants labeled E in both the Jan07 and Mar18 meetings are meeting managers who are responsible for organizing the meeting. Clearly, E in Jan07 plays an active role in keeping the meeting on track; this can be observed by simply viewing the meeting video but also from the basic floor statistics. E in the Jan07 meetings speaks the greatest number of words and backchannels the most. However, E in the Mar18 meeting plays a more "nominal" meeting manager role. From the basic statistics of the meeting, we observe that C speaks more words than E, and G has the most backchannels. Given the special role played by an active meeting manager, we analyzed whether the meeting manager affects floor change even when he/she is not a previous or next floor holder. In the Jan07 meeting, there were 53 cases that E is not either the previous or next floor holder in floor exchange (only *Change* and *Overlap*). In these 53 cases, E gazes at the next floor holder 21 times. However, sometimes the manager gazes to the next holder together with other participants. If we rule out such cases, E gazes to the next floor holder 11 times (20.75%). This suggests that an active meeting manager's gaze target plays some role in predicting the next floor holders. In the Mar18 meeting, there are 100 cases that E is not a floor holder. For these 100 cases, E gazes to the next floor holder only 6 times. In fact, it is the case that E gazes largely at his papers or the whiteboard.

Gesture Events: Gesture has been found to help coordinate floor control. During a floor transitions, the previous floor holder may point to a meeting participant to assign the floor. When a person desires to gain control of the floor, he/she may use hand movements, such as lifting their hand or some object (e.g., pen) in accompaniment with speech in order to attract attention to gain the floor. Here we consider whether gestural cues could be helpful for an automatic floor control detection system. We calculated the number of occurrences of *floor giving* (**fg**) gestures used by the previous floor holder and the *floor capturing* (**fc**) gestures used by the next floor holder during floor transitions in the two meetings. As can be seen in Figure 6, there are many more floor capturing gestures in these two VACE meetings than floor giving gestures. Therefore, in an automatic floor control prediction system, concurrent floor capturing gestures should provide useful cues from the gesture domain.

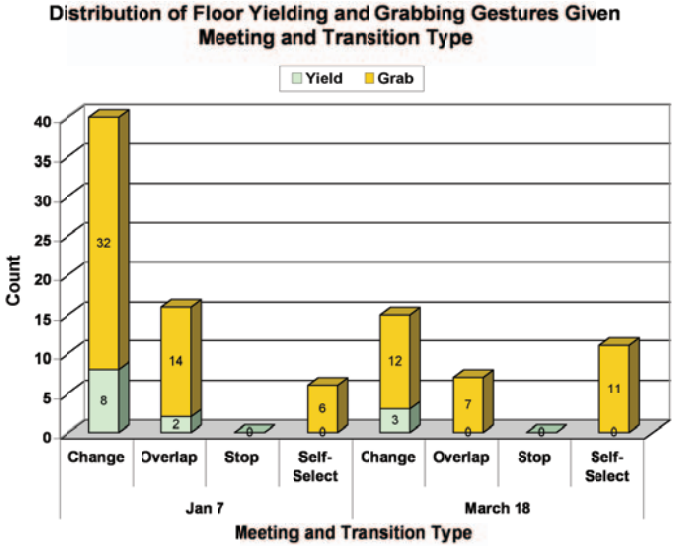


Fig. 6. Gestures for grabbing and yielding the floor

3 Conclusions

The floor control structure of a meeting provides important information for understanding that meeting. We presented a floor control annotation specification and applied it to two different meetings from the VACE meeting corpus. From an analysis of these markups, we have identified some multimodal cues that should be helpful for predicting floor control events. Discourse markers are found to occur frequently at the beginning of a floor. During floor transitions, the previous holder often gazes at the next floor holder and vice verse. The well-known mutual gaze break pattern in dyadic conversations is also found in the Jan07

meeting. A special participant, an active meeting manager, is found to play a role in floor transitions. Gesture cues are also found to play a role, especially with respect to floor capturing gestures. Comparing the Jan07 and Mar18 meetings, we find that implements (e.g., papers and white boards) in the meeting room environment impact participant behavior. It is important to understand the factors that impact the presence of various cues based on the analysis of a greater variety of meetings.

In future work, we will refine our floor control annotation specification and continue to annotate more meetings in the VACE collection, as well as in other meeting resources. Using knowledge obtained from these measurement studies, we will build an automatic floor control prediction system using multimodal cues.

Acknowledgements

We thank all of our team members for their efforts in producing the VACE meeting corpus: Dr. Yingen Xiong, Bing Fang, and Dulan Wathugala from Virginia Tech, Dr. David McNeill, Dr. Susan Duncan, Jim Goss, Fey Parrill, and Haleema Welji from University of Chicago, Dr. Ron Tuttle, David Bunker, Jim Walker, Kevin Pope, Jeff Sitler from AFIT. This research has been supported by ARDA under contract number MDA904-03-C-1788 and by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings, and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of ARDA and DARPA.

References

[1] M. Argyle and M. Cook. *Gaze and Mutual Gaze.* Cambridge Univ. Press, 1976.
[2] L. Bosch, N. Oostdijk, and J. P. Ruiter. Durational aspects of turn-taking in spontaneous face-to-face and telephone dialogues. In *Proc. of TSD 2004*, pages 563–570, 2004.
[3] S. Burger, V. MacLaren, and H. Yu. The ISL meeting corpus: The impact of meeting type on speech type. In *Proc. of Int. Conf. on Spoken Language Processing (ICSLP)*, 2002.
[4] J. Cassell, J. Sullivan, S. Prevost, and E. Churchill. *Embodied Conversational Agents.* MIT Press.
[5] L. Chen, T. Rose, F. Parrill, X. Han, J. Tu, Z. Huang, I. Kimbara, H. Welji, M. Harper, F. Quek, D. McNeill, S. Duncan, R. Tuttle, and T. Huang. VACE multimodal meeting corpus. In *Proceeding of MLMI 2005 Workshop*, 2005.
[6] S. Duncan. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23:283–292, 1972.
[7] C. E. Ford and S. A. Thompson. Interactional units in conversation: syntactic, intonational, and pragmatic resources for the management of turns. In T. Ochs, Schegloff, editor, *Interaction and Grammar.* Cambridge Univ. Press, 1996.
[8] J. Garofolo, C. Laprum, M. Michel, V. Stanford, and E. Tabassi. The NIST Meeting Room Pilot Corpus. In *Proc. of Language Resource and Evaluation Conference*, 2004.

[9] N. Jovanovic and R. Akker. Towards automatic addressee identification in multi-party dialogues. In *Proceedings of SIGDial*, 2004.
[10] N. Jovanovic, R. Akker, and A. Nijholt. A corpus for studying addressing behavior in multi-party dialogues. In *Proc. of SIGdial Workshop on Discourse and Dialogue*, 2005.
[11] D. Jurafsky, B. Rebecca, and et al. Automatic detection of discourse structure for speech recognition and understanding. In *Proc. of IEEE Workshop on Speech Recognition and Understanding*, 1997.
[12] M. Kipp. Anvil: A generic annotation tool for multimodal dialogue. In *Proc. of European Conf. on Speech Processing (EuroSpeech)*, 2001.
[13] Y. Liu. *Structural Event Detection for Rich Transcription of Speech.* PhD thesis, Purdue University, 2004.
[14] J. Local and J. Kelly. Projection and 'silences': Notes on phonetic and conversational structure. *Human Studies*, 9:185–204, 1986.
[15] I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(3):305–317, 2005.
[16] N. Morgan and et al. Meetings about meetings: Research at ICSI on speech in multiparty conversations. In *Proc. of ICASSP*, volume 4, pages 740–743, Hong Kong, Hong Kong, 2003.
[17] D. G. Novick, B. Hansen, and K. Ward. Coordinating turn-taking with gaze. In *Proc. of Int. Conf. on Spoken Language Processing (ICSLP)*, 1996.
[18] E. Padilha and J. Carletta. A simulation of small group discussion. In *Proc. of the sixth Workshop on the Semantics and Pragmatics of Dialogue (EDILOG 2002)*, pages 117–124, Edinburgh, UK, 2002.
[19] E. Padilha and J. Carletta. Nonverbal behaviours improving a simulation of small group discussion. In *Proceedings of the First International Nordic Symposium of Multi-modal Communication*, 2003.
[20] T. Rose, F. Quek, and Y. Shi. MacVisSTA: A system for multimodal analysis. In *Proc. of Int. Conf. on Multimodal Interface (ICMI)*, 2004.
[21] H. Sacks, E. Schegloff, and G. Jefferson. A simplest systematics for the organisation of turn taking for conversation. *Language*, 50:696–735, 1974.
[22] E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proc. of SIGdial Workshop on Discourse and Dialogue*, 2004.
[23] O. Torres, J. Cassell, and S. Prevost. Modeling gaze behavior as a function of discourse structure. In *Proc. of the First International Workshop on Human-Computer Conversations*, Bellagio, Italy, 1997.
[24] K. Weilhammer and S. Rabold. Durational aspects in turn taking. In *International Congresses of Phonetic Sciences*, 2003.
[25] A. Wichmann and J. Caspers. Melodic cues to turn-taking in English: Evidence from perception. In *Proc. of SIGdial Workshop on Discourse and Dialogue*, 2001.

Combining User Modeling and Machine Learning to Predict Users' Multimodal Integration Patterns

Xiao Huang[1], Sharon Oviatt[1,2], and Rebecca Lunsford[1,2]

[1] Natural Interaction Systems
10260 Sw Greenburg Road Suite 400
Portland, OR 97223
{Xiao.Huang,Rebecca.Lunsford}@naturalinteraction.com
[2] Center for Human-Computer Communication
Computer Science Department
Oregon Health and Science University
Beaverton, OR 97006
oviatt@csee.ogi.edu

Abstract. Temporal as well as semantic constraints on fusion are at the heart of multimodal system processing. The goal of the present work is to develop *user-adaptive temporal thresholds* with improved performance characteristics over state-of-the-art fixed ones, which can be accomplished by leveraging both empirical user modeling and machine learning techniques to handle the large individual differences in users' multimodal integration patterns. Using simple Naïve Bayes learning methods and a leave-one-out training strategy, our model correctly predicted 88% of users' mixed speech and pen signal input as either unimodal or multimodal, and 91% of their multimodal input as either sequentially or simultaneously integrated. In addition to predicting a user's multimodal pattern in advance of receiving input, predictive accuracies also were evaluated after the first signal's end-point detection─the earliest time when a speech/pen multimodal system makes a decision regarding fusion. This *system-centered metric* yielded accuracies of 90% and 92%, respectively, for classification of unimodal/multimodal and sequential/simultaneous input patterns. In addition, empirical modeling revealed a .92 correlation between users' multimodal integration pattern and their likelihood of interacting multimodally, which may have accounted for the superior learning obtained with training over heterogeneous user data rather than data partitioned by user subtype. Finally, in large part due to guidance from user-modeling, the techniques reported here required as little as 15 samples to predict a "surprise" user's input patterns.

1 Introduction

Techniques for temporal information fusion are at the heart of designing a new generation of multimodal systems. Current state-of-the-art multimodal systems use *fixed temporal thresholds* based on previous modeling of users' natural modality integration patterns [1,2]. However, newer studies [3,4,5] show that there are significant individual differences among users in their multimodal integration patterns, and that adaptive temporal thresholds for multimodal systems could achieve substantial

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 50 – 62, 2006.

improvements in processing speed, accuracy and overall performance [6,7]. Motivated by these newer results, our recent work addresses the development of user-adaptive temporal thresholds for future multimodal systems. The present paper explores combining empirical user modeling and machine learning techniques to quickly learn a user's multimodal integration patterns, and to adapt a multimodal system's temporal thresholds to that user.

1.1 Related Work on Individual Differences in Multimodal Integration Patterns

A series of studies conducted with users across the lifespan has indicated that individual child, adult, and elderly users all adopt either a predominantly *simultaneous* or *sequential* integration pattern during production of speech and pen multimodal constructions [3,4,5,7]. It can be summarized that: 1) previous lifespan data on speech and pen input from over 100 users shows that they are classifiable as either *simultaneous* or *sequential* multimodal integrators (70% simultaneous, 30% sequential); 2) a user's dominant simultaneous or sequential integration pattern can be identified almost immediately; and 3) their integration pattern remains highly consistent throughout an given interaction (88-97% consistent) and over time. 4) Based on previous data, it's clear that users' dominant multimodal integration pattern is strikingly consistent and resistant to change. In addition, behavioral and linguistic differences in the interaction styles of these two groups suggest underlying enduring differences in cognitive style [7].

Based on previous research, Table 1 summarizes the multimodal input ratio of ten adults while interacting with a map-based multimodal system [5,7,8]. Participants interacted multimodally on 62% of the tasks and unimodally on 38%, and there were large individual differences in the ratio of multimodal interaction ranging from 22% to 92%. Given hand annotated data, previous research has indicated that a human rater could predict both a user's dominant multimodal integration pattern (i.e., simultaneous/sequential) and their likelihood of interacting multimodally (i.e., versus unimodally) with 100% accuracy after only 15 commands [8]. All of these findings indicate

Table 1. Average percentage of unimodal vs. multimodal interactions, and sequential vs. simultaneous integration patterns for different user's multimodal interactions ([8])

Subject	Multimodal	Unimodal	SIM	SEQ
1	69%	31%	87%	13%
2	92%	8%	100%	0%
3	62%	38%	90%	10%
4	62%	38%	97%	3%
5	84%	16%	99%	1%
6	89%	11%	98%	2%
7	22%	78%	5%	95%
8	69%	31%	72%	28%
9	41%	59%	97%	3%
10	28%	72%	0%	100%
Consistency	73.6%		93.5%	

that users' multimodal interaction and integration patterns are fertile content for incorporating machine-learning techniques. Future multimodal systems that can *detect and adapt to a user's dominant multimodal integration patterns* could yield substantial improvements in multimodal system robustness and overall performance.

1.2 Related Work Applying Machine Learning to Multimodal Data

In order to build adaptive temporal thresholds, a multimodal system has to be able to learn and adapt to each user's input patterns. However, the general study of adaptive information fusion for multimodal systems is still in its infancy [5]. Apart from standard stream-weighting techniques for optimizing multimodal signal recognition, more recent work has begun investigating and developing new machine learning techniques in areas like adaptive information fusion for audio-visual speech processing, user authentication, and activity classification [9, 10, 11, 12]. Typically, such research uses graphical models (Hidden Markov Models or Bayesian Belief Networks and their extensions) to build models of the relation between different modalities.

For example, Bengio [9, 12] proposed an asynchronous Hidden Markov Model for audio-visual speech recognition and user authentication. This work takes advantage of the inherently close "temporal coupling" of speech and lip movements as modalities, and it requires a large amount of high-quality video and acoustic training data. For example, after training conducted with 185 recordings from 37 subjects, performance with the AHMM exceeded that of an HMM (i.e., yielding 88.6% correct for 9 digits at 10 dB signal-to-noise ratio). In the case of data on users' speech and pen input, these modes are not as closely aligned temporally, and sometimes do not occur in combination at all. As such, this is a more challenging problem than processing closely-coupled multimodal data. One major under-acknowledged prerequisite for processing multimodal speech and pen input is *accurate clustering of users' speech and pen signals into multimodal versus unimodal constructions* before fusion and semantic interpretation take place, which is one goal of the present research.

In other work by Oliver, layered HMMs [10] have been used to infer 6 distinct human activities in an office environment from users' audio-visual activities and mouse input after 1 hour of training. With respect to data requirements, Oliver's methods were more efficient than most others outlined in the literature (i.e., 10 mins. of training for each of 6 activities). High accuracy also was reported for activity classification (over 99%), although generalization across variations in office conditions (e.g., changes in lighting) is known to be a limitation with this type of approach.

In work conducted by Lester, et. al [11], static classifier and HMM models were combined to predict users' physical activities with a wide variety of multimodal sensors (e.g., auditory activity, acceleration) while people were mobile. After training on 4 hours of data at a frequency of 4 Hz, mobile users' activities could be identified with 85% accuracy. In summary, most previous learning models have required relatively large amounts of training data. In contrast, one goal of our current work is to develop accurate learning models for predicting users' multimodal interaction patterns after *minimal training samples* (i.e., as few as 15 samples total, learned over 1-3 mins.), such that they can be deployed easily during *real-time multimodal processing*.

In earlier work on the development of adaptive multimodal processing techniques for handling users' integration patterns, Gupta developed adaptive temporal

thresholds for fusion based on BBNs, which was implemented within a speech/pen multimodal system. In this work, he reported a 40% performance improvement after training on 495 samples, compared with systems that use fixed temporal thresholds [6]. Apart from empirical user modeling in this area (cited in section 1.1), in past work we also have implemented simple Bayesian Belief Network models with discrete variables, which achieved prediction accuracies of 85% in classifying users' multimodal integration patterns after only 15 training samples [8].

1.3 Why Combine Empirical User Modeling and Machine Learning Techniques?

It is well known that in many cases machine-learning techniques [13,14] (e.g., HMMs, Neural Networks) can be computationally intractable unless one has prior knowledge to bootstrap machine-learning models, which is one of our motivations for combining empirical user modeling and machine learning techniques. In addition, other advantages of using empirical modeling to guide machine learning applications include that it can indicate: 1) what content is most fertile for applying learning techniques; 2) what gains can be expected if learning techniques are applied; and 3) when different learning techniques should be applied to handle different subgroups of users adequately. Instead of selecting information sources through trial and error, user modeling also can 4) guide the selection of information sources; 5) indicate how to apply learning techniques so they are transparent and avoid destabilizing users' performance; and 6) reveal how many training samples are needed to train a learning model to achieve a given level of performance. If a model requires too many training samples, it may be inappropriate for real-time learning and/or may not be fertile territory for applying machine learning techniques to certain real-world problems.

1.4 Specific Goals of This Research

In this paper, we combine user modeling and machine learning techniques in an effort to predict users' multimodal integration patterns. Our goals include: (1) conducting further empirical work to discover what type of information may best predict users' multimodal input patterns, which then could be leveraged as prior knowledge to bootstrap machine learning, and (2) determining the best training strategy for improving the predictive accuracy and speed of learning users' multimodal input patterns. With respect to the second goal, we investigated a) the impact of training sample size to discover the optimal amount of training data, b) training over each user's data, data partitioned by user types, and training over all (heterogeneous) user data together, and c) the efficiency of the leave-one-out train-test technique with the present multimodal data, which also can evaluate how well a multimodal system can adapt to a new "surprise" user's input patterns.

Finally, we wanted to develop and test a predictive model that could be used during real-time multimodal system's decision-making regarding whether to fuse input signals before attempting lexical interpretation. Rather than predicting the type of user input completely in advance of receiving it, we evaluated the model's predictive power after end-point detection of a first signal, which is the earliest time at which a speech/pen multimodal system would attempt signal fusion and interpretation.

2 Empirical Study on Users' Multimodal Interaction Patterns

2.1 Study Overview

Data used in this research were collected from 10 volunteer users while interacting spontaneously with a multimodal map-based system. During practice, participants completed 5 tasks using speech only, 5 using pen only, and 5 using both speech and pen. After training, participants were told they could interact with the system any way they wished for the remainder of the session, using speech input, pen input, or multimodal input. To ensure there was no effect of recognition-based system errors on the users' interaction choices, a high-fidelity Wizard-of-Oz system was employed with errors generated at a fixed rate of 20% for all conditions. For further details, see [5].

Input from participants was coded as either unimodally or multimodally delivered. If unimodal, input was scored as either involving speech input or pen input. When multimodal, the integration pattern was coded as either simultaneous (i.e., speech and pen input at least partially overlapped in time), or a sequential one (i.e., one input mode delivered before the other, with a lag between modes). Each participant also was classified as having a dominant unimodal/multimodal and simultaneous/sequential pattern if 60% or more of their input could be classified as that type. An independent second scorer carefully double-checked all of the real-time unimodal and multimodal judgments and multimodal integration patterns to verify their accuracy.

2.2 Results: Correlated Multimodal Integration Patterns

Participants' ratio of simultaneous versus sequential multimodal integrations was strongly correlated to their ratio of multimodal versus unimodal interactions, p=0.92. In fact, 85% of the variance in a participant's likelihood of generating a multimodal versus unimodal construction could be accounted for just by know

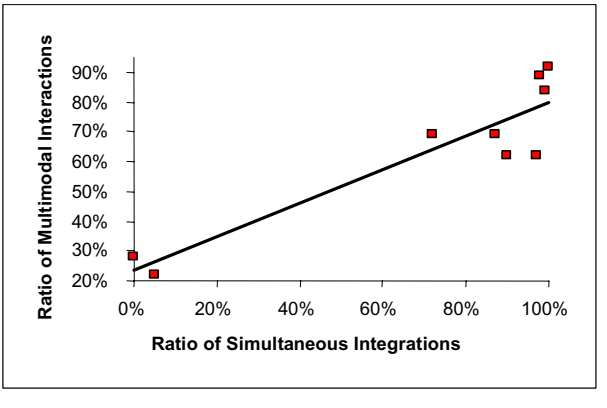


Fig. 1. Linear regression between participants' ratio of simultaneous to sequential multimodal integrations, and their overall likelihood of interacting multimodally

ing their ratio of simultaneous to sequential multimodal integrations during multimodal interactions, which was significant, F=38.3 (df=1,7) $p<.00005$, two-tailed. Figure 1 shows the best fitting linear regression, with the ratio of multimodal interactions increasing in relation to a participant's ratio of simultaneous integrations.

2.3 Discussion

This correlation between participants' ratio of simultaneous to sequential integrations and their likelihood of interacting multimodally is very substantial. Given that users' multimodal integration patterns were highly consistent, this finding provides powerful predictive leverage on correctly classifying a user's subsequent signal patterns. In fact, knowing a user's dominant integration pattern can account for 85% of all the variance in their likelihood of interacting multimodally during subsequent input. In future work, it remains an open question how best to leverage this strong correlation to bootstrap optimal predictive power, although two possibilities are (1) incorporation of more relevant information source into new models, and (2) pursuit of heterogeneous training during machine learning which, given adequate modeling, would be a prerequisite for detecting the regularities between these correlated signal patterns.

3 Machine Learning Approaches for Input Pattern Prediction

In this section, we first provide an introduction to Bayesian Belief Networks and its simplified version, Naïve Bayes. We then compare and present the results of three different training strategies. The general goal was to investigate how best to combine user modeling and machine learning techniques to build a new generation of adaptive multimodal interfaces. More specifically, our results provide guidance for developing user-adaptive temporal thresholds for fusion in future multimodal systems.

3.1 Introduction to Bayesian Belief Network and Naïve Bayes

A Bayesian Belief Network (BBN) [15] is a graphical model that encodes probabilistic relations among discrete related variables. A BBN model can infer causal relations and handle situations where some data are limited or missing. Furthermore, it also is an ideal representation for combining prior knowledge and new training samples.

Naïve Bayes models, a simplified version of BBN, are simple to implement and efficient to train and use, typically producing reasonable predictions compared with more complex learning-based models. However, by assuming that variables are independent and equally important, they also can cause skewed results, especially if many of the variables are interrelated. Because of the ease of implementing Naïve Bayes, we chose this model as a starting point for the present exploratory work even though the multimodal information sources we are modeling are known to be interrelated.

We used the Matlab toolkit [16] to implement a Naïve Bayes model (Figure 2) for this study. The model represents the joint probability distribution of seven variables (four input, three output): 1) Type of current signal: an input variable that represents the type of the modality represented in the current signal (speech, pen, or neither/silence); 2) Duration of current signal: an input variable that has two values, 1 if the duration is longer than the average duration, and 0 if less; 3) Last multimodal

integration pattern: an input variable value of the last multimodal integration pattern (simultaneous, sequential or neither if a unimodal interaction); 4) Last command type: an input variable of the last interaction's unimodal/multimodal value; 5) Type of next signal: an output variable that represents what the new next signal is (i.e., if the interaction is unimodal, the next signal would be silence); 6) Command type: an output variable that represents whether the interaction is unimodal or multimodal; 7) Multimodal integration pattern: an output variable that represents the predicted temporal relationship between the current signal and next signal.

We selected these variables during initial modeling for three reasons. First, they are available and fully annotated in the dataset. Second, they are either discrete or can be rendered as discrete variables, which is compatible with constraints entailed in building discrete BBN models. Third, they represent basic signal and command-level information sources, which are good candidates for initially attempting to predict users' command type and integration pattern.

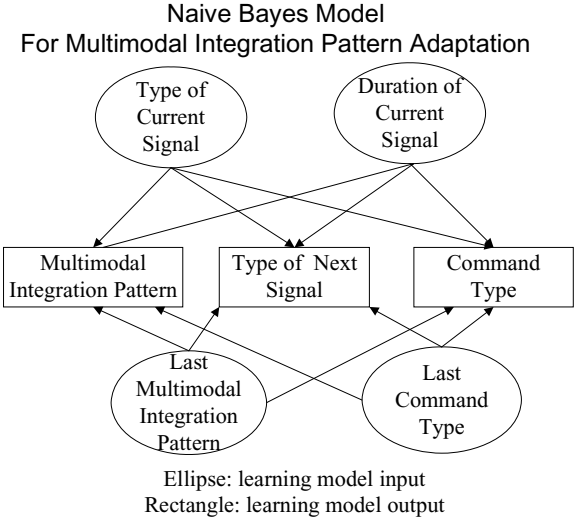


Fig. 2. Machine Learning Model (Naïve Bayes)

3.2 Tests of Machine Learning Approaches

As mentioned earlier, one goal of this work was to investigate means for combining user modeling with machine learning techniques to improve the speed, accuracy and generalizability of predicting users' command type and integration pattern. Towards that end, we investigated different training strategies to determine the best learning models and strategies for predicting users' signal input.

For the first test, we used different size training sets, increasing the number of training samples from 5 to 10, 15, 30, and 45, with the goal of determining how few samples are enough to train a user-adaptive learning model that can adequately predict users' multimodal input patterns. Determining the optimal training sample size

is important in order to know if a given domain is a candidate for online learning and also to avoid overtraining. If a learning model needs too many training samples, then real-time online learning could become intractable.

The second test was to partition train/test sets according to the two main types of user interaction pattern (i.e. unimodal or multimodal), and to determine whether these input patterns which represent different user groups may benefit from applying different learning techniques. As a result, a Naïve Bayes model was built for each user subset. The goal was to examine whether partitioning based on prior user-modeling knowledge could bootstrap the accuracy of prediction yielded by machine-learning.

The third test involved applying the leave-one-out technique, which is a typical strategy for organizing training and test subsets of data during evaluation of machine learning methods. We divided the entire dataset into two subsets: a set including the data from one subject (A) and another set containing data from the rest of the subjects (B). Set B was defined as the training set, while set A was the test set. Training and test data were recomputed 10 times for each of the 10 subjects in this manner, and then averaged. Unlike the partitioning during test 2, this average predictive accuracy represented training across the full heterogeneous group of all diverse users. To the extent the model shown in Figure 2 incorporates information sources involved in the correlated signal patterns reported in section 2 (i.e., between users' multimodal integration pattern and their likelihood of interacting multimodally), then predictive accuracy would be expected to improve with training over the more heterogeneous data involved in this third test, in comparison with user-partitioned training in test 2.

3.3 System-Centered Evaluation Metric of Machine Learning

In addition to predicting a given user's multimodal signal pattern *in advance of receiving a construction,* which was the learning metric compared during the first three tests, predictive accuracies also were evaluated during a fourth test *after the first signal's end-point detection*— which is the earliest time when a speech/pen multimodal system needs to make a decision regarding fusion. This *system-centered metric* was developed because we also need learning models that are capable of real-time prediction and decision-making about whether to complete lexical interpretation of an incoming signal after detecting its' end-point— or to wait and fuse the signal with a later arriving one before interpreting their joint meaning. Therefore, in this test we assumed that the model knows the end-point of the user's first signal. If the input pattern is multimodal and simultaneous, then readiness for lexical processing is clear and there is no need for prediction. Otherwise, the system needs to decide whether the present signal is unimodal, or multimodal but a part of sequentially-integrated construction. Instead of using a fixed temporal threshold which requires waiting 2-4 seconds before resolving this ambiguity, a system with a user-adaptive temporal threshold can weight the likelihood that an upcoming construction is unimodal or multimodal based on previous history. With this system-centered processing viewpoint in mind, a fourth machine learning test was conducted based on a new model.

4 Machine Learning Results

4.1 Increasing the Number of Training Samples

In this experiment, we built a Naïve Bayes model for each subject. The number of training samples varied from 5 to 10, 15, 30 and 45. The number of testing sample was fixed in all cases at 38. As shown in Figure 3, the average prediction accuracies for unimodal/multimodal and simultaneous/sequential for 5 and 10 samples were relatively low (5 samples: 64% and 58%; 10 samples: 74% and 68%). In contrast, using 15 samples, the performance improved substantially (79% and 81%). Further increasing the number of training samples provided minimal improvement beyond this (30 samples: 78% and 79%; 45 samples: 85% and 82%). These results are consistent with previous empirical results [7], in which using the first 15 samples for each user was sufficient to provide optimal classification of users' dominant input patterns.

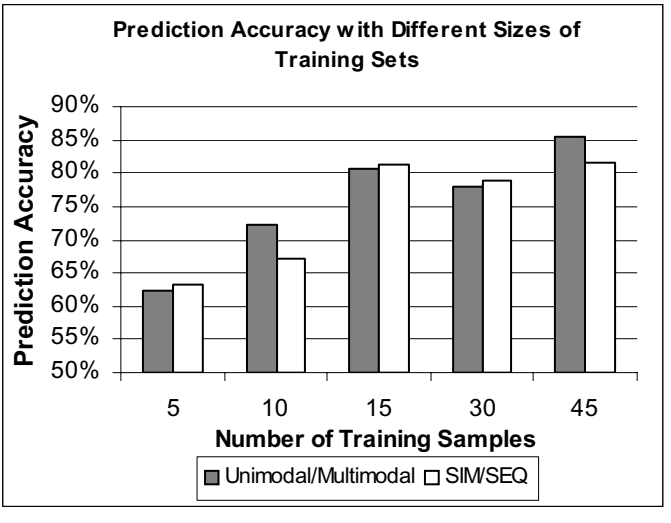


Fig. 3. Prediction accuracies of unimodal/multimodal and simultaneous/sequential patterns, with training sample sizes varying from 5, 10, 15, 30 to 45, respectively

4.2 Partitioning Training into Unimodal/Multimodal User Subsets

In this experiment, we partitioned the dataset into two user subsets. Set 1 includes all habitually unimodal subjects. Set 2 includes the multimodal subjects. We built a Naïve Bayes model for each subset, and compared the results with a model built for each individual subject. For Set 1, there are 45 training samples (i.e., 15 training samples from each of 3 unimodal subjects) and 214 test samples (i.e., 68 from each of 3 subjects). For Set 2, there are 105 training samples and 476 test samples (i.e., based on 7 subjects total). We also conducted a "Baseline" experiment by building a learning model for each individual subject (15 training samples and 68 test samples), yielding 10 total. The average prediction accuracies for unimodal/multimodal and simultaneous/sequential were 79.4% and 81.3%, respectively, for the baseline model. With the

data partitioned by user subtype, the average prediction accuracies were 83.5% and 77.9%, respectively, which was similar to accuracy of the baseline model.

4.3 Leave-One-Out Test Method

In this experiment, we used the last 68 samples from a given user as the test set and the first 15 samples from the other users (135 total samples) as the training set, and then repeated the procedure 10 times, once for each subject. Using this training strategy, 88% of users' natural mixed input could be correctly classified as either unimodal or multimodal, and 91% of users' multimodal input could be correctly classified as either sequentially or simultaneously integrated, as shown in Figure 4. These high predictive accuracies exceeded the results achieved after partitioning training by user subtypes. This performance level may have derived in part from training across heterogeneous user data, which permitted learning of the strong correlation between multimodal information sources outlined in section 2.

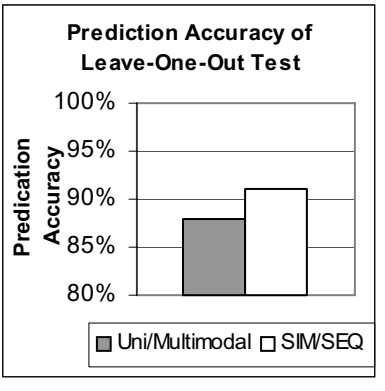


Fig. 4. Prediction accuracy for classifying unimodal/multimodal and simultaneous/sequential patterns based on the leave-one-out training strategy

4.4 Learning Methods Applied to System-Centered Fusion Process

The "Type of Current Signal" variable used in the previous model had three possible values: speech, pen and silence. In this experiment, we assumed the model is applied after the end-point of the first signal, and that the "Type of Current Signal" variable has one more value—"Both signals" (i.e., co-occurring). For this evaluation, we built a learning model for each subject. The first 15 samples were used for training, and the remaining 68 samples for testing. Using this model, 90% of users' natural mixed input could be correctly classified as either unimodal or multimodal, and 92% of their multimodal input was correctly classified as either sequentially or simultaneously integrated, as shown in Figure 5. These high accuracies indicate that real-time systems could be very effectively guided by user-adaptive predictions during the actual process of fusion and lexical interpretation.

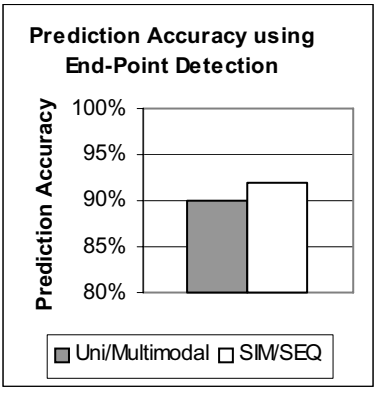


Fig. 5. Prediction accuracy for classifying unimodal/multimodal and simultaneous/sequential patterns based on end-point detection

5 Discussion and Future Work

In this paper, we combined user modeling with machine learning approaches to leverage better prediction of users' multimodal integration patterns. Motivated by previous empirical results, we investigated three different training strategies and two separate metrics of machine learning performance. Using just 15 training samples for each subject, the Naïve Bayes learning model achieved 79% prediction accuracy for unimodal/multimodal classification of users' input, and 81% accuracy for users' simultaneous versus sequential multimodal constructions. Increasing the number of training samples beyond this did not further enhance prediction accuracy. This result is consistent with past empirical studies, in which users' dominant patterns could be classified by humans with 100% accuracy after 15 samples, based on hand annotations. Secondly, we divided the training data into two user subgroups based on each subject's dominant multimodal interaction pattern (unimodal vs. multimodal). However, predictive accuracies based on partitioned data did not exceed rates achieved by training on more heterogeneous combined data during the leave-one-out test. In the third leave-one-out test, the machine learning model correctly classified 88% of users' natural mixed input as either unimodal or multimodal, and 91% of users' multimodal input as either sequentially or simultaneously integrated. These high predictive accuracies may have been due in part to the fact that this model was trained on hetero geneous user patterns, which would have enabled learning of the high correlation between information sources that was summarized on section 2. Finally, system-centered modeling that involved prediction after end-point detection of the first signal revealed accuracies for classifying unimodal/multimodal input of 90%, with classification of simultaneous/sequential multimodal integrations at 92%. These high accuracy rates based on a simple Naïve Bayes approach create a promising basis for developing a new generation of multimodal systems with adaptive temporal thresholds.

The long-term goal of this research is automatic learning and real-time system adaptation to users' multimodal integration patterns, as well as the development of new strategies for combining empirical user modeling with machine learning techniques to

bootstrap the accelerated, generalized, & improved reliability of information fusion in new types of multimodal systems─ including ones involving different modalities and applications. Based on this work, it is clear that empirical user modeling can guide machine learning techniques by uncovering fertile applications and valuable information sources. It also can provide insights into why machine learning succeeds when it does, which will be valuable for generalizing machine learning techniques successfully. Future work will need to explore the performance of more sophisticated learning models, such as asynchronous HMM models [12] and Markov Logical Networks [17] at handling this type of multimodal integration data. In addition, future work should develop learning models based on more precise continuous temporal information, so that users' average signal overlap or lag can be predicted more precisely.

Acknowledgments

Thanks to Benfang Xiao and Josh Flanders for assistance with data collection. This research was supported by DARPA Contract No. NBCHD030010 and NSF Grant No. IIS-0117868. Any opinions, findings or conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Project Agency, or the Department of the Interior.

References

1. S. Oviatt. Ten myths of multimodal interaction. Comm. of the ACM, Vol. 42(11). (1999) 74–81
2. S. Oviatt. Integration and synchronization of input modes during multimodal human computer interaction. In: Proc. of CHI. (1997) 415–422
3. S. Oviatt, R. Coulston, S. Tomko, B. Xiao, R. Lunsford, M. Wesson, and L. Carmichael, Toward a theory of organized multimodal integration patterns during human-computer interaction. In: Proc. of ICMI. (2003) 44-51
4. B. Xiao, R. Lunsford, R. Coulston, M. Wesson, and S. Oviatt. Modeling multimodal integration patterns and performance in seniors: Toward adaptive processing of individual differences. In: Proc. of ICMI. (2003) 265–272
5. S. Oviatt, R. Coulston, and R. Lunsford. When do we interact multimodally? Cognitive load and multimodal communication patterns. In: Proc. of ICMI. (2004) 129-136
6. A. Gupta and T. Anastasakos, Dynamic time windows for multimodal input fusion, In: Proc. of Interspeech. (2004) 2293-2296
7. S. Oviatt, R. Lunsford and R. Coulston: Individual differences in multimodal integration patterns: What are they and why do they exist? In: Proc. of CHI. (2005) 241-249
8. X. Huang and S. Oviatt, Towards adaptive information fusion in multimodal systems, In: Proc. of MLMI. (2005) 15-27
9. S. Bengio. An asynchronous hidden Markov model for audio-visual speech recognition. In: Proc. of Advances in Neural Information Processing Systems. (2003) 1213–1220
10. N. Oliver, A. Garg and E. Horvitz, Layered representations for learning and inferring office activity from multiple sensory channels, Int. Journal on Computer Vision and Image Understanding, 96(2) (2004) 163-180
11. J. Lester, T. Choudhury and G. Borriello, A Practical approach to recognizing physical activities, To appear in the Proc. of Pervasive. (2006)

12. S. Bengio. Multimodal authentication using asynchronous HMMs. In: Proc. of AVBPA. (2003) 770–777
13. L. Rabiner, A tutorial on hidden Markov model and selected applications in speech recognition. In: Proc. of the IEEE, Vol.77, No.2 (1989) 257-286
14. R. Duda, P. Hart and D. Stork, Pattern classification, Morgan Kaufmann (2002)
15. D. Heckerman. A tutorial on learning with Bayesian networks. Learning in Graphical Modals. MIT Press (1999)
16. K. Murphy. The Bayes net toolbox for Matlab. Computing Science and Statistics, Vol. 33. (2001)
17. M. Richardson and P. Domingos, Markov Logic Networks, Machine Learning, Vol. 62. (2006) 107-136

Using Audio, Visual, and Lexical Features in a Multi-modal Virtual Meeting Director

Marc Al-Hames, Benedikt Hörnler,
Christoph Scheuermann, and Gerhard Rigoll*

Institute for Human-Machine-Communication, Technische Universität München
Arcisstr. 21, 80290 Munich, Germany
{alh,hbe,chr,rigoll}@mmk.ei.tum.de

Abstract. Multi-modal recordings of meetings provide the basis for meeting browsing and for remote meetings. However it is often not useful to store or transmit all visual channels. In this work we show how a virtual meeting director selects one of seven possible video modes. We then present several audio, visual, and lexical features for a virtual director. In an experimental section we evaluate the features, their influence on the camera selection, and the properties of the generated video stream. The chosen features all allow a real- or near real-time processing and can therefore not only be applied to offline browsing, but also for a remote meeting assistant.

1 Introduction

Projects like Augmented Multi-Party Interaction (AMI) [2, 3], Computers in the Human Interaction Loop (CHIL) [8], or the ICSI meeting project [4] investigate how computers and machine learning techniques can be used to make meetings, lectures, and conferences more efficient, and how to automatically record, transcribe, analyse, and summarise them. One goal of the AMI project is the development of a meeting browser [10] that allows to recapitulate a meeting from its audio-visual recordings and automatically generated transcripts. Furthermore in an international world and with the required technology (like web-cams) now cheaply available, remote meetings are of emerging importance. These meetings allow people to connect audio-visual from their own office to other meeting rooms or participants. This allows regular meetings, while saving travel costs and especially the time of the meeting participants.

Both a meeting browser and remote meetings require to select one video stream, that is shown either to the user of the meeting browser, or to a remote participant. We can of course always show a merged visual stream with all meeting participants. Yet this is not desirable, because watching all persons at the same time is not convenient. Furthermore with an increasing number of

* This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811).

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 63–74, 2006.

participants all subtle details get lost, but they can contain important information, like disagreement [2]. Thus for both a meeting browser and remote meeting transmissions, a selection which camera should be shown is required.

One could possibly argue, that the task of a virtual director is in fact a speaker diarisation task: simply always select the camera that shows the current speaker. However, imagine a person presenting a novel idea in a meeting with the project board. The presentation lasts for about five minutes. If the virtual director follows a speaker diarisation rule, it will show the presenter all the time. Unfortunately this way the system has lost the most important aspect of the idea: during the presentation the project leader has continuously shook his head, indicating he is not very satisfied. This important moment is lost if only the acoustic channel is considered. Meetings are truly multi-modal in nature [1], important information can be in a camera view that doesn't correspondent to the current speaker. This concept is followed by directors of TV talk-shows: They often show persons not currently talking. They wait for their reactions like gestures, or facial expressions. A virtual director has to take care of this as well. Thus selecting a camera is not a speaker diarisation, but a multi-modal task.

In this work we investigate different audio, visual, and lexical features for a virtual director. We introduce seven video modes and explain how they can be derived from the recordings in the smart meeting room. We'll discuss their advantages and when they are best shown in the meeting. We will then show how simple, yet very useful features for meeting analysis can be derived. While simple, they can be derived in real-time, which is important for online analysis. They are not only useful for a virtual director, but also for other kinds of meeting analysis (points of interest, individual and group actions). We will give measures how good the features are, and what the individual strengths, or weaknesses are.

2 Meeting Data

The meeting data for this work has been collected in the AMI and the M4 project [3]. The AMI project uses three different meeting rooms. In this work we concentrate only on the meetings recorded in the IDIAP smart meeting room. The room is equipped with various recording devices (as described below), a table, a whiteboard, and a projector with screen. A schematic of this room and three sample camera outputs are shown in Fig. 2. Each meeting has four participants (P1 - P4).

Close-talking audio is recorded with an omni-directional lapel and a headset condenser microphones for each participant (in Fig. 2 the microphones are indicated by black dots). Far-field recordings are performed with two microphone arrays: A1 is placed on the table in the middle of the participants and consists of eight miniature omni-directional electret microphones. The second array A2 with four microphones is mounted on the ceiling. Furthermore the room is equipped with a binaural manikin (BM) for two further recordings.

Video is recorded with seven cameras: four cameras record closeup views (1 - 4) of the meeting participants. Two cameras record a left (L), resp. right (R)

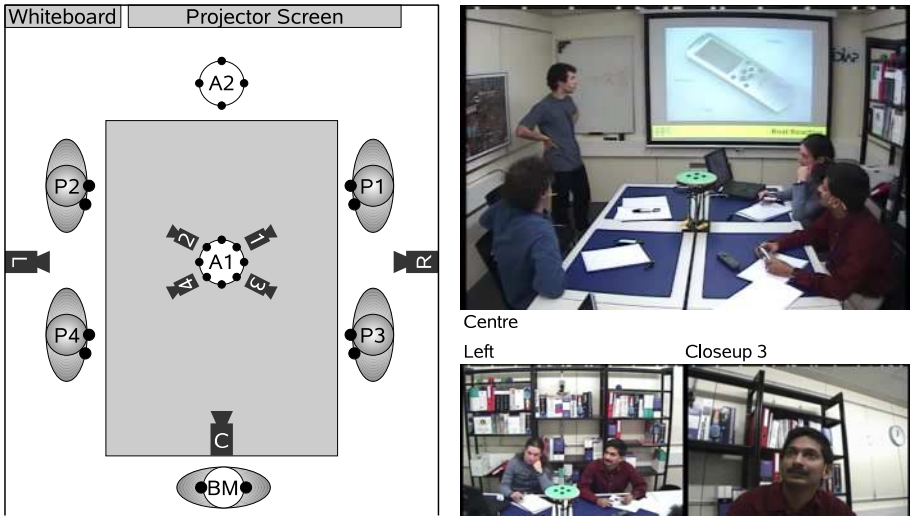


Fig. 1. Schematic of the AMI IDIAP smart meeting room (not drawn to scale)

view of the room; each showing two participants and the table in front of them. The last camera (C) captures a total of the room with all four participants, the table, as well as the whiteboard, and the projector screen.

The content of the projection board is recorded with time-stamps as a series of static images. Furthermore the whiteboard is captured as x-y-coordinates of the pen and individual notes with Logitech I/O digital pens.

All recordings in the room are time-synchronised with a central timecode. In this work we use the the lapel and headset microphones and all visual recordings from 41 videos with different lengths.

3 Video Modes

The task of a virtual meeting director is to select for each frame one camera or one view from the available cameras. In the case of a remote meeting, this view is then transmitted, for later browsing the selected view is stored. Based on the available seven cameras in the meeting room and the possible user requirement we defined seven different video modes. They are shown in Fig. 3 and shall be described shortly:

Mode 1 (P1-P4): Shows the closeup camera of one of the persons P1 - P4. This is the main mode when a person is talking or shows facial expressions.

Mode 2: Shows the left-camera view and thus the persons P1 and P3. This mode is ideal for a discussion between the two, or as a diversification if P1 or P3 talks (a stylistic device that human directors often use in talk shows). It can also be used if P1 or P3 talks, and the other one reacts in some way – e.g. a shaking of the head.

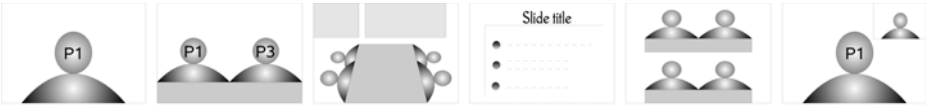


Fig. 2. Available video modes for the virtual director: from left to right 1, 2, 4, 5, 6, and 7. Mode 3 is omitted, as it is analogues to mode 2, but uses the right camera.

Mode 3: Shows the right-camera and thus the persons P2 and P4, it corresponds to mode 2 and has the same properties.

Mode 4: Shows a total of the room from the central camera. This total involves the whiteboard, the projection board, and all four participants. It is ideal if somebody gives a presentation, or to show group interactions. However the individual persons are rather small in this mode. Furthermore the persons are shown from the side, thus details get lost.

Mode 5: This mode inserts a still image (slides, pictures, etc.) into the video. It is ideal to show up the presentation slides when they are changed.

Mode 6: Shows both the output of the left and the right camera. They are slightly cut on top and the bottom, scaled down, and then merged on top of each other. This mode shows all participants in a frontal view and is therefore good for group discussions, note-taking, or group interactions. The individual persons are larger and better shown as in mode 4, but due to the adding up of two views smaller than in mode 1, 2, and 3. Thus individual reactions are less impressive. Furthermore the cutting contains the risk of cutting out heads or hands.

Mode 7 (P1-P4, P1-P4): Shows the closeup camera of one of the persons P1 - P4. A further closeup of another person is merged into the corner. This view can be used to show reactions of one participant, while another person is talking. However if the persons are sitting next to each other, mode 2 or 3 are preferred, as mode 7 is rather unnatural.

The presented modes are of course adapted to the conditions of the smart meeting room. However similar modes showing the same sets of group, or individual dynamics can easily be derived for other meeting room settings. The presented features in this work are not limited to the here presented modes and can in principle be adapted to different requirements or other meeting rooms.

4 Features

4.1 Visual Features

As a first visual feature we use global motions (GM). They have been successfully applied to various meeting tasks [13, 9] and can be calculated in real-time. We first split the smart meeting room into six locations L. Each of the four closeup cameras represents one location. From the centre view camera we extract the

projection board and the whiteboard location. Then a difference image sequence $I_d^L(x,y)$ is calculated for each of these six locations by subtracting the pixel values of two subsequent frames from the video stream. Then the seven global motion features are derived from the image sequence, again for each location. The centre of motion is calculated for the x- and y-direction according to:

$$m_x^L(t) = \frac{\sum_{(x,y)} x \cdot |I_d^L(x,y,t)|}{\sum_{(x,y)} |I_d^L(x,y,t)|} \quad \text{and} \quad m_y^L(t) = \frac{\sum_{(x,y)} y \cdot |I_d^L(x,y,t)|}{\sum_{(x,y)} |I_d^L(x,y,t)|}. \tag{1}$$

The changes in motion are used to express the dynamics of movements:

$$\Delta m_x^L(t) = m_x^L(t) - m_x^L(t-1) \quad \text{and} \quad \Delta m_y^L(t) = m_y^L(t) - m_y^L(t-1). \tag{2}$$

Furthermore the mean absolute deviation of the pixels relative to the centre of motion is computed:

$$\sigma_x^L(t) = \frac{\sum_{(x,y)} |I_d^L(x,y,t)| \cdot \left(x - m_x^L(t)\right)}{\sum_{(x,y)} |I_d^L(x,y,t)|}$$

and

$$\sigma_y^L(t) = \frac{\sum_{(x,y)} |I_d^L(x,y,t)| \cdot \left(y - m_y^L(t)\right)}{\sum_{(x,y)} |I_d^L(x,y,t)|}. \tag{3}$$

Finally the intensity of motion is calculated from the average absolute value of the motion distribution:

$$i^L(t) = \frac{\sum_{(x,y)} |I_d^L(x,y,t)|}{\sum_x \sum_y 1}. \tag{4}$$

These seven features are concatenated for each time step in the location dependent motion vector

$$\boldsymbol{x}^L(t) = [m_x^L, m_y^L, \Delta m_x^L, \Delta m_y^L, \sigma_x^L, \sigma_y^L, i^L]^T. \tag{5}$$

With this motion vector the high dimensional video stream is reduced to a seven dimensional vector, but it preserves the major characteristics of the currently observed motion. Concatenating the motion vectors from each of the six positions $\boldsymbol{x}^L(t)$ leads to the final motion vector $\boldsymbol{x}_V(t) = [\boldsymbol{x}^{C_1}, \boldsymbol{x}^{C_2}, \boldsymbol{x}^{C_3}, \boldsymbol{x}^{C_4}, \boldsymbol{x}^W, \boldsymbol{x}^P]^T$, that describes the overall motion in the meeting room with 42 features.

4.2 Head and Hand Blobs

While the global motion features of the six locations in the meeting room provide a fast and simple access to the location dependent activity, they only reflect the participants' motions in a very compressed, summarised way. A better way to access the individual participants activities are hand and head movements. In [5]

it was shown how hand and head skin blobs can be used to detect the activity of individual meeting participants. We therefore add skin blobs as a visual feature.

In this work we extract the head and hand skin blobs with a simple - yet in the context of recorded meetings successful - skin colour look up table approach[11]. First the RGB-images are transformed into the rg-space

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}, \tag{6}$$

which is less crucial to light changes. Each image pixel is then compared to a 16 bit rg-look up table. This table has been learned from 5.7 Million skin colour pixels from different non-meeting images. The comparison with the skin colour table results in a binary image, where each possible skin pixel is marked. Then a 5x5 dilation filter [6] is applied to the binary image. This filtering leads to an extension of skin pixel areas. Especially gaps in large skin pixel areas are filled with this filter. The found skin pixel areas are then analysed for their shape and for the relation of their eigenvalues. Only blobs that are large enough for a face, resp. hands, are selected; these selected areas are further narrowed down by the relation between the blob edges. Furthermore we apply context knowledge about the usual position of heads and hands of meeting participants. This already leads to a quite stable finding of heads and hands in the meeting videos. Then subsequent images are averaged by a recursive approach, that is applied individually to all hand and head blobs in the meeting videos:

$$\boldsymbol{m}(t) = 1 - \frac{1}{T}\boldsymbol{m}(t-1) + \frac{1}{T}\boldsymbol{x}(t), \tag{7}$$

where $\boldsymbol{x}(t)$ is the current measured value, $\boldsymbol{m}(t)$ is the resulting averages vector for the blob position, $\boldsymbol{m}(t-1)$ the position in the last image, and T a constant that determines the relation between previous frames and the current measurement.

This approach is simple, yet fast and for the target application reliable enough. Of course it would be better to apply a face and hand tracking. Once these systems are available in the AMI project we will exchange this module with advanced tracking methods for meeting scenarios, as e.g. compared in [7]. This however only improves the accuracy of the coordinates, the principles suggested in this work and the influence of this feature to virtual editing will not be influenced.

4.3 Acoustic Features

Beside the visual features we derive a range of simple acoustic meeting features. In this work we use the information from each of the four participants lapel microphones. At first we use a frame energy derived directly from the audio stream from each microphone. While simple, this approach is very fast and gives at least a cue for a speaker activity detection.

As second acoustic feature we perform a short time fast Fourier transform of the microphone channels. We then filter this short time spectrum with a band-path filter to extract only regions relevant for speech. The filtered values are then

summed up, resulting in a "frequency frame energy" that highlights the spoken aspects in the microphone channels. Again, this approach is simple and can easily be performed in real-time. In the future we plan to introduce a further speaker diarisation module from the AMI project [2], which should better discriminate between speech and non-speech regions. However the drawback of these methods is that they often can not be applied in real time. The frequency frame energy method is therefore required for remote meetings, where real-time processing is necessary.

As the third acoustic feature we derive 13 Mel frequency cepstral coefficients, and their first and second derivations. They will later be used in the statistic fusion process, but are not applied in this work. In the future we also plan to use the position dependent microphone array information.

4.4 Lexical Features

We also use higher semantic information as input to the virtual meeting director. Group actions in meetings have been deeply investigated [1]. We revert to the eight well-known group action classes:

Monologue (person 1-4): One person speaks without being interrupted.
Discussion: Two or more persons talk alternately.
Presentation: One person gives a presentation in front of the projection board.
Whiteboard: One person writes on the whiteboard and talks.
Note-taking: (All) persons write something down.

Sometimes the discussion class is further split into disagreement and consensus to reflect the kind of discussion. This is however not required for a virtual director. To better model the interaction between the group and individuals, it has also been suggested to combine different meeting group actions [12], note-taking with either monologue, presentation, or whiteboard. This combinations will not be used in this work.

The meeting group actions have two main advantages, making them very well suited for automatic video editing: they can very reliable detected from the raw-audio visual stream (see [1] for a comparison of various automatic recognition models). And compared to other semantic information (like dialogue acts), the meeting group actions can easily be mapped to video modes, e.g. a monologue of a participant to a camera view of this particular person. There is one drawback of this feature: while in principle possible, the automatic recognition models do usually not work in real-time. The use of the feature for remote meetings is therefore currently not possible. However for an offline editing and later browsing, they can be applied. And with emerging computer power, and the models at hand, online detection of meeting group actions will become possible in the future.

5 Feature Scopes

In the last section we showed a range of features that can be applied to virtual meeting editing. We have derived these features on a per frame basis. If this

output is directly used as an input to a virtual director, this can lead to unintentional twitches. We therefore also evaluate the influence of windowing a range of subsequent frames for the audio and the visual features. We apply three different window functions, where each function represents a different feature scope: a history oriented, a history-future balanced, and a future oriented approach.

In the following let T denote the total length of the meeting, t the current time step, $P \in \{P_1, P_2, P_3, P_4\}$ one of the meeting participants, W the window size (with increasing W the scope of the features is increased), and $F^P(t)$ be the addressed feature for person P at time t. Then the windowed output of the feature is denoted as $D^P(t)$. The *history scope* sums up features from the history:

$$D_h^P(t) = \sum_{\tau=0}^{t} F^P(\tau) \qquad \text{for } t < W, \tag{8}$$

$$D_h^P(t) = \sum_{\tau=t-W}^{t} F^P(\tau) \qquad \text{for } t >= W. \tag{9}$$

The windowed output $D_h^P(t)$ therefore represents what has recently happened. It does not reflect what will happen in the near future. In contradiction, the *balanced scope* sums up features from the history and the near future:

$$D_b^P(t) = \sum_{\tau=0}^{t+W/2} F^P(\tau) \qquad \text{for } t < W/2, \tag{10}$$

$$D_b^P(t) = \sum_{\tau=t-W/2}^{t+W/2} F^P(\tau) \qquad \text{for } W/2 <= t <= T - W/2, \tag{11}$$

$$D_b^P(t) = \sum_{\tau=t-W/2}^{T} F^P(\tau) \qquad \text{for } t > T - W/2. \tag{12}$$

The windowed output $D_b^P(t)$ represents both what has recently happened and what will happen in the near future. Finally the *future scope* sums up features from the future only:

$$D_f^P(t) = \sum_{\tau=t}^{t+W} F^P(\tau) \qquad \text{for } t <= T - W, \tag{13}$$

$$D_f^P(t) = \sum_{\tau=t}^{T} F^P(\tau) \qquad \text{for } t > T - W. \tag{14}$$

The windowed output $D_f^P(t)$ therefore represents only what will happen in the near future, without including past actions.

Only the history oriented approach is causal and can therefore be applied to online processing. The other two concepts can only be applied offline. However in the experimental section we will show that they are indeed useful for browsing.

6 Video Mode Decision

The scope of this work is to investigate the influence of the different features to a virtual meeting director. We therefore do not apply statistical or other machine learning methods for the actual video mode decision. For the acoustic and the visual features we calculate the windowed output $D^P(t)$ for each time and each person. We then choose the "most active" person with

$$V(t) = \underset{P}{\operatorname{argmax}}\, D^P(t). \tag{15}$$

Depending on the desired output this decision $V(t)$ can now be mapped directly to one of the seven video modes (e.g. an activity of person $P2$ to mode 1). For the semantic group action features a decision function is not required, they can directly be mapped to a video mode (e.g. a discussion to mode 2).

This process of course does not optimise the features, nor does it model interactions between the features. This way the influence of the features to the video output stream is not diluted and can therefore best be evaluated. In future the features and the evaluation experience can then be used for statistical models (like HMMs).

7 Experiments

We performed three sets of experiments: In the first we evaluate the single features, both on a per frame basis and windowed. In a second experiment we perform a simple fusion scheme. In the last experiment we investigate the influence of the three different window scopes history, balanced, and future.

The output of a virtual director can not be trivially evaluated with "objective measures". Indeed directing is depending on taste, the difference between the cut of two human directors can be huge. However there are some measures, that indicate the quality of the feature. As a first evaluation measure we use the percentage similarity to the output of the meeting group actions. That is, we compare the output of the audio and visual features with the output of the meeting group action features. As the meeting group action features model the interactions between the participants, this is a meaningful number, how strong the feature represents interactions. However, a high value does not automatically correspond to a good visual stream (or this work would be reduced to the problem of meeting group action recognition). As a second value we use the average frequency of mode changes. This value indicates the number of cuts in the stream and thus corresponds to how fast the system changes to activities in other channels. We also evaluate the maximum segment length per meeting: this value shows the longest time period without a cut. A very low number indicates that the system is not staying in one view for a sufficient period. On the other hand a very long segment can be rather boring to watch.

Finally we give a user rating score between zero and ten (where ten is the best) for each feature. While this is not an objective measure, it expresses the significance of the individual features from a user's point of view.

Table 1. Evaluation results for the lexical, acoustic, and visual features: Similarity to the meeting group actions, average mode changes per minute, and maximum length of a sequence without mode changes in seconds. Standard deviations are given in brackets.

Feature	*Group Similar.*	*Changes/Min*	*Max. length/s*	*Score*
Group Actions	100.0 (0.0)	1.1 (0.3)	116.0 (31.3)	4
Audio (per Frame)	21.7 (13.8)	373.3 (83.4)	6.1 (4.3)	0
Audio (History)	25.0 (15.7)	18.7 (8.3)	83.0 (37.0)	5
Frequency (per Frame)	20.5 (12.9)	435.7 (78.0)	3.6 (1.2)	0
Frequency (History)	25.0 (15.7)	16.5 (8.1)	81.6 (37.2)	6
Global Motion (per Frame)	20.8 (9.3)	205.5 (42.9)	7.9 (5.2)	0
Global Motion (History)	19.6 (10.2)	32.1 (5.3)	19.8 (13.8)	4
Skin Blobs (per Frame)	13.5 (6.1)	935.4 (85.6)	0.9 (0.3)	0
Skin Blobs (History)	14.3 (10.4)	63.8 (18.6)	16.1 (9.9)	2

7.1 Feature Results

We first evaluated the influence of the features to the virtual editing. The results are presented in Tab. 1. Each audio and visual feature was both evaluated on a per frame and with history basis. The group actions have a very low change frequency. Once a speaker gives a monologue, they do not change at all, thus information in other channels can get lost. However they are very stable to watch. On the other hand all "per frame" features are – if used directly – not watchable at all, they twitch heavily between the modes. In the case of the blobs a mode change happens in average in every second frame, the longest segment has duration of less than a second. Such a cut of a meeting can't be watched.

These results get much better if the history is taken into account, according to (8) and (9). Then the frequency of mode changes reduces for all four features to reasonable numbers, e.g. for the audio feature to 18.7 changes per minute. In general it can be seen that the two visual features result in more changes than the acoustic features. Especially the blobs still have a very high mode change.

In a second experiment we applied a very simple late fusion scheme, where the single feature video modes are merged with a confidence multiplication. The results after fusion for different feature combinations are shown in Tab. 2. It can be seen, that this simple late fusion scheme doesn't lead to better video outputs. The high frequency of mode changes remains unchanged. This directly calls for advanced statistical methods to benefit from the information in all channels, but on the other hand reduce the high mode changes.

7.2 Feature Scope Evaluation

In the last experiment we investigated the influence of the three different window functions (8) - (14). The *history scope* is the only causal scheme, and therefore the only one that can be applied to a remote meeting, as this requires online processing. The drawback of the history scope is it's lack to react to actions

Table 2. Evaluation results after late fusion of different feature combinations. If not otherwise indicated, the methods are with history and not on a per frame basis.

Feature	*Group Similar.*	*Changes/Min*	*Max. length/s*	*Score*
All (per Frame)	20.6 (10.9)	641.5 (112.7)	3.2 (1.8)	0
All (History)	25.6 (14.2)	26.6 (9.2)	51.2 (22.1)	5
Audio + Frequency	25.0 (15.7)	16.5 (8.1)	81.6 (37.2)	6
Global Motion + Skin Blobs	14.3 (10.4)	63.8 (18.6)	16.1 (9.9)	2
Global Motion + Audio	26.1 (13.9)	22.2 (6.5)	48.4 (19.9)	4
Skin Blobs + Audio	23.7 (14.0)	38.3 (13.0)	41.6 (26.1)	4

in the future. If, e.g. a participant shakes his head, the history scope can only react after the shaking has started. A human director can of course neither know what will happen in the next seconds, but a human director can use intuition and from time show other participants. Therefore if the history scope is used for online processing a virtual director has to model some "intuition" and from time to time randomly switch to other channels.

In our evaluations the *balanced scope* gave the best results. We suggest to use it for offline meeting editing. It both reacts on past activities, and switches fast enough to observe the start of reactions. The pure *future scope* is interesting to watch, as all user reactions are fully captured. However from time to time the system tends to switch to fast to a different participant and then nothing happens for a while. Thus the balanced scope is preferred.

8 Conclusion

In this work we introduced five features for a virtual meeting director. The presented features are all very simple, but useful and - especially important for remote meetings - can in principle be processed online in real-time. We deeply investigated the influence of the different features on the resulting video stream. We showed that acoustic features alone are not sufficient for this task, but visual features are required. We also investigated a simple late fusion scheme and the influence of different window scopes.

We now have a measurement where the strength and weakness of the individual features are. In the future we plan to model the group and individual interactions with statistical models. This should result in a much more advanced and better virtual meeting director. The features and evaluation results from this work will be of valuable input to such a system.

References

[1] M. Al-Hames, A. Dielmann, D. Gatica-Perez, S. Reiter, S. Renals, G. Rigoll, and D. Zhang. Multimodal integration for meeting group action segmentation and recognition. In S. Renals and S. Bengio, editors, *MLMI 2005, 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*. Springer Verlag, 2006.

[2] M. Al-Hames, T. Hain, J. Cernocky, S. Schreiber, M. Poel, R. Muller, S. Marcel, D. van Leeuwen, J.M. Odobez, S. Ba, H. Bourlard, F. Cardinaux, D. Gatica-Perez, A. Janin, P. Motlicek, S. Reiter, S. Renals, J. van Rest, R. Rienks, G. Rigoll, K. Smith, A. Thean, and P. Zemcik. Audio-visual processing in meetings: Seven questions and current AMI answers. In *MLMI 2006, 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*. Springer Verlag, 2006.

[3] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI meetings corpus. In *Proceedings of the Measuring Behavior 2005 symposium on Annotating and measuring Meeting Behavior*, 2005.

[4] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The ICSI meeting corpus. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003.

[5] I. Potucek, S. Sumec, and M. Spanel. Participant activity detection by hands and face movement tracking in the meeting room. In *Proceedings IEEE Computer Graphics International (CGI)*, pages 632–635, 2004.

[6] W.K. Pratt. *Digital image processing*. John Wiley & Sons, 2001.

[7] K. Smith, S. Schreiber, V. Beran, I. Potúcek, and D. Gatica-Perez. A comparitive study of head tracking methods. In *MLMI 2006, 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*. Springer Verlag, 2006.

[8] A. Waibel, H. Steusloff, R. Stiefelhagen, and the CHIL Project Consortium. CHIL: Computers in the human interaction loop. In *Proceedings of the NIST ICASSP Meeting Recognition Workshop*, 2004.

[9] F. Wallhoff, M. Zobl, and G. Rigoll. Action segmentation and recognition in meeting room scenarios. In *Proceedings IEEE International Conference on Image Processing (ICIP)*, Singapore, October 2004.

[10] P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with Ferret. In S. Bengio and H. Bourlard, editors, *Machine Learning for Multimodal Interaction: First International Workshop, MLMI 2004*. Springer Verlag, 2005.

[11] M.-H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transasctions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[12] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, and G. Lathoud. Modeling individual and group actions in meetings: a two-layer hmm framework. In *Proceedings IEEE Workshop on Event Mining at the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[13] M. Zobl, F. Wallhoff, and G. Rigoll. Action recognition in meeting scenarios using global motion features. In J. Ferryman, editor, *Proceedings Fourth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS)*, pages 32–36, 2003.

A Study on Visual Focus of Attention Recognition from Head Pose in a Meeting Room

Sileye O. Ba and Jean-Marc Odobez

IDIAP Research Institute, Martigny, Switzerland

Abstract. This paper presents a study on the recognition of the visual focus of attention (VFOA) of meeting participants based on their head pose. Contrarily to previous studies on the topic, in our set-up, the potential VFOA of people is not restricted to other meeting participants only, but includes environmental targets (table, slide screen). This has two consequences. Firstly, this increases the number of possible ambiguities in identifying the VFOA from the head pose. Secondly, due to our particular set-up, the identification of the VFOA from head pose can not rely on an incomplete representation of the pose (the pan), but requests the knowledge of the full head pointing information (pan and tilt). In this paper, using a corpus of 8 meetings of 8 minutes on average, featuring 4 persons involved in the discussion of statements projected on a slide screen, we analyze the above issues by evaluating, through numerical performance measures, the recognition of the VFOA from head pose information obtained either using a magnetic sensor device (the ground truth) or a vision based tracking system (head pose estimates). The results clearly show that in complex but realistic situations, it is quite optimistic to believe that the recognition of the VFOA can solely be based on the head pose, as some previous studies had suggested.

1 Introduction

An important aspect of human being daily life, as social being, is interaction with other humans. A topic of intense study in psychology is the ways these interactions happen in groups such as families or work teams [10]. Human interactions happen through speech or non verbal cues. One one hand, the use of verbal cues in groups is rather well defined because it is tightly connected to to the taught explicit rules of language (grammar, dialog acts). On the other hand, the usage of the non verbal cues is usually more implicit, which does not prevent it from following rules and exhibiting specific patterns in conversations. A person rising hand often means that he/she is requesting a speaking turn. In face to face conversation, a listener's head nod or shake can be interpreted as agreement or disagreement [6]. Besides hand and head gestures, the visual attention of people, as defined by the eye gaze, is another important cue of non verbal communication. For instance, gaze is often used as a mean to regulate the dialog. Speaker's gaze is related to back-channel request, whereas listener's gaze can be used to request for speaking turns [4,11]. Furthermore, studies have shown that a person's visual attention was influenced by the visual attention

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 75–87, 2006.

state of other people [7]. Thus, in brief, recognizing the visual attention pattern of group of people can reveal an important amount of information about the social nature of the occuring interactions and help us to better understand them. Due to its importance, after psychologists, computer vision researchers are currently investigating the identification and role of the gaze in social interaction in situations such as meetings in smarts rooms, making use of all the multi-modal abilities of such environments [17,18].

As part of our effort to study human interaction, the goal of this paper is to analyze the correspondence between the head pose and the eye gaze of people. In other words we want to evaluate how well we can infer the visual focus of attention (VFOA) solely from the head pose. This study, conducted in the context of a smart meeting room environment, complements previous research in this domain. First our study generalizes to more complex situations similar works that have already been conducted in [12,17]. Contrarily to these previous works, the scenario we consider involves people looking at slides or writing on the table. As a consequence, in our set-up, people have more potential visual focus of attention (6 instead of 3 in [12,17]). Also, due to the physical spatial configuration of the VFOA targets, the identification of the VFOA can only be done using complete head pose representation (pan and tilt), instead of just the head pan as done previously. Finally, in our set-up, there were ambiguities between VFOA depending on people's sitting location in the room while in the previous work there were not. Thus our study reflects more complex, but realistic, meeting room situations in which people don't just focus their attention on the other people but also on other targets such as the table, the white board, the slide screen, or even sometimes are not focused on any predefined object.

In this paper, we propose to analyze the recognition of the VFOA of people from their head pose. In our experiments, the head poses are either obtain using a magnetic sensor (the ground truth) or a computer vision based probabilistic tracker, allowing to evaluate the degradation in VFOA recognition when going from true values to estimated ones. VFOA are then recognized using either the Maximum A Posteriori principle or an Hidden Markov Models (HMM) modeling, where in both cases the VFOAs are represented using Gaussian distributions. Our early experimental results, based on VFOA frame and event performance measures, show that in the case of more complex environments, previous work results were probably optimistic in concluding that it was possible to infer the VFOA from head pose alone, or, to a lower extent, to use tracking estimates instead of ground-truth head pose measurements.

The remaining of this paper is organized as follows. Section 2 discusses works related to ours. Section 3 indicates the way we obtain head pose measures using either a magnetic field sensor, or our probabilistic method for joint head tracking and pose estimation, and present a numerical evaluation of the latter one. Section 4 describes the considered models for recognizing the VFOA from head pose. In Section 5, we give experimental results and Section 6 concludes the paper.

2 Related Work

The estimation of the VFOA of a person from visual input has been studied in the past using different approaches. For instance, in applications studying the visual exploration of images by people, wearable eye gaze trackers are often used. Wearable sensor for eye gaze tracking are infrared based systems. An infrared light is shined on the subject whose gaze is tracked, which creates a red eye effect. The difference of reflexion between the cornea and the pupil is then exploited to determine the gaze direction. As an example, [13] studied people's attentions and reactions to advertisement exposure using such technology, to understand where advertiser should put important information to capture clients' attention. However, besides the concerns over the safety of long exposure to infrared lights, wearable sensor can be used only in controlled experimental situations. In other situations, non invasive procedures to estimate the eye gaze are required. This is the case for instance of systems which aim at automatically detecting driver (loss of) attention. In such cases, computer vision technics can be used, given the availability of high resolution images, to estimate the gaze direction of a driver. In a representative example, [15], motion and skin color distribution are used to track facial features and from the eye balls location reconstruct the gaze direction. In the human computer interaction domain, [9], a similar approach has been used to estimate the gaze location of a worker siting in front of his computer in an office environment.

Although eye gaze tracking with computer vision technics is less invasive than eye gaze tracking with a wearable sensor, it is still relatively constraining, as the subject usually has to remain close to the camera because tracking eye features requires high resolution images. Thus, some people proposed to estimate the VFOA from head pose instead of eye gaze. As a good example, [17] showed that, in a 4 people meeting configuration, the hypothesis that the head is approximately oriented in the same direction as the gaze is a reasonable assumption. In this work, however, there was no ambiguity between the head poses which were defining people's different VFOA, because of their specific meeting physical set-up (four participants evenly space around a round table). Also, the head poses were assimilated to the head azimuth (head pan) only. Following [17], other researchers used the same assumption regarding head pose and eye gaze to model the VFOA of people. For instance, in a very interesting work, [12] makes use of both people's utterance information and head azimuth angle, obtained from a magnetic field head orientation sensor, to infer conversational models in a 4 persons conversation. In [3], the head pose was exploited to model visual attention in office from which workers social geometry was defined, where the social geometry defines when people are available or not for communication.

3 Head Pose Tracking

For our study we used a database comprising 8 meetings of 4 people (duration of meetings ranged from 7 to 14 minutes), recorded in IDIAP's smart meeting

Fig. 1. Left: sample image of the dataset; Right: visual focus of attention targets

room. The scenario of the meeting was to discuss statements displayed on the projection screen. A sample image of the data is shown in Figure 1. Due to technological constraints[1], we were able to capture the head ground truth of only two participants (the left and right person in Fig. 1), using 3D magnetic sensors attached to the head. The head pose is defined as an instance of head rotation with respect to a reference configuration. In general, head poses are represented by three Euler angles (α, β, γ) which parameterize the decomposition of the rotation matrix of the head configuration with respect to the camera frame. Among the possible decompositions, we have selected the one whose rotation axes are rigidly attached to the head. With this choice, we have: α denotes the pan angle, a left/right head rotation; β denotes the tilt angle, an up/down head rotation; and finally, γ, the roll, represents a left/right "head on shoulder" head rotation (see Figure 2). In the following we describe the two alternative techniques that we used to extract the head pose information. The first one consisted of using orientation sensors, and the obtained head poses defined the ground truth values. The second approach provides head pose estimates based on a computer vision probabilistic tracking algorithm.

3.1 Head Pose Ground Truth from Magnetic Sensors

The head pose ground truth was obtained using a 3D location and orientation magnetic sensor called flock of bird (FOB) [19]. The coordinate frame of this sensor was calibrated with respect to the camera frame. In each recording, the time delay between the FOB and the video was set by detecting the occurrence of the same events (peak oscillations) in both modalities.

3.2 Probabilistic Method for Head Pose Tracking

Here, we summarize the approach we employed and which is described in [1].

The probabilistic framework for tracking is well known. Denoting by X_t the hidden state representing the object configuration at time t and by Y_t an observation extracted from the image, the objective is to estimate the filtering distribution $p(X_t|Y_{1:t})$ of X_t given all the observations $Y_{1:t}$. In non-Gaussian and

[1] The magnetic field due to the setup caused distortions on the flock-of-birds readings.

Table 1. Pointing vector, pan, tilt and roll errors statistics

pointing vector			pan			tilt			roll		
mean	std	med	mean	std	med	mean	std	med	mean	std	med
20.3	11.3	18.2	9.10	8.6	7.0	17.6	12.2	15.8	10.1	9.9	7.5

non linear cases, this can be done recursively using sampling approaches, also known as particle filters. The idea behind particle filter consist in representing the filtering distribution using a set of weighted samples $\{X_t^n, w_t^n, n = 1, ..., N_s\}$ and updating this representation when new data arrive. Given the particle set of the previous time step, $\{X_{t-1}^n, w_{t-1}^n, n = 1, ..., N_s\}$, configurations of the current step are drawn from a proposal distribution $X_t \sim \sum_n w_{t-1}^n p(X|X_{t-1})$. The weights are then computed as $w_t \propto p(Y_t|X_t)$. Four elements are important in defining a particle filter: i) a state model defining the object we are interested in ii) dynamical model governing the temporal evolution of the state $p(X_t|X_{t-1})$ iii) a likelihood model measuring the adequacy of data given the proposed configuration of the tracked object and iv) a sampling mechanism which have to propose new configurations in high likelihood regions of the state space. These elements along with our model are described in the next paragraphs.

State Space: The state space contains both continuous and discrete variables. More precisely, the state is defined as $X = (S, \theta, l)$ where S represent the head location and size, θ represents the head in-plane rotation. Both S and θ parameterize a transform $\mathcal{T}_{S,\theta}$ defining the head spatial configuration. The variable l label an element of the discretized set of possible head poses.

Dynamical Model: The dynamical model governing the temporal evolution of the state is defined as

$$p(X_t|X_{1:t-1}) = p(\theta_t|\theta_{t-1}, l_t)p(l_t|l_{t-1}, S_t)p(S_t|S_{t-1}, S_{t-1}) \quad (1)$$

The dynamics of the head in plane rotation variable θ_t and the discrete head pose variable l_t are learned using the head pose GT training data. Head location and size variables dynamics are modelled as a second order auto-regressive process.

Observation Model: The observations $Y = (Y^{text}, Y^{col})$ are composed of textures and color observations. Assuming that, given the state value, texture and color observation are independent, the observation likelihood was modeled as:

$$p(Y|X = (S, \theta, l)) = p_{text}(Y^{text}(S, \theta)|l)p_{col}(Y^{col}(S, \theta)|l) \quad (2)$$

where for each head pose variable l, the parameters of the texture likelihood p_{text} and the color likelihood p_{col} were learned from the Prima-Pointing database [5] containing head image appearances of the pose l. For a given hypothesized configuration X, the parameters (S, θ) allow to extract an image patch, on which the features are computed, while the examplar index l allows to select the appropriate appearence model.

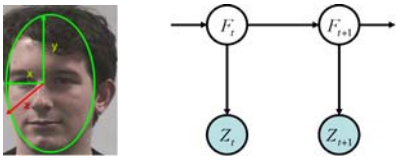


Fig. 2. Left: basis attached to the head (head pointing vector in red). Right: visual focus of attention graphical model.

Sampling Method: In this work, we use Rao-Blackwellization which consist in applying the standard PF algorithm over the tracking variables S and γ while applying an exact filtering step over the exemplar variable l. The method theoretically results in a reduced estimation variance, as well as a reduction of the number of samples.

3.3 Head Pose Tracking Evaluation

For evaluation, we followed the protocol described below.

Data: we used the IHPD database[2]. Amongst the 16 recorded people with ground truth, we used half of the database (8 people) as training set to learn the pose dynamic model and the half remaining as test set to evaluate the tracking algorithms. Because of the scenario used to record data, people often have negative pan values corresponding to looking at the projection screen. The pan values range from -70 to 60 degree. Tilt values range from -60 (when people are writing) to 15 degrees, and roll value from -30 to 30 degrees.

Performance measures: four error measures are used. The three first measures are the errors in pan, tilt and roll angle, i.e. the average of the absolute difference between the pan, tilt and roll of the ground truth (GT) and the tracker estimation. The fourth error measure is defined by the average angular error between the 3D pointing vector (the unit vector of the z-axis of the basis attached to the head cf Figure 2) defined by the head pose GT and the pose estimated by the tracker. Note that this vector depends only on the head pan and tilt values (given the selected representation, cf first paragraph of Section 3).

Results: The statistics of the errors over the test set are showed in Table 1. Overall, given the small head size, and the fact that the appearence training set is composed of heads recorded in an external set up (and thus does not contain any individuals from our testing database), the results are quite good, with a majority of head pan errors smaller than 10 degrees. However, these results hide a large discrepancy between individuals. For instance, the average pan error ranges from 4 degrees to 15 degrees, and depends mainly on whether the tracked person resembles one of the person of the training set used to learn the appearance model. The table also shows that the errors in pan and roll are smaller than the errors in tilt. This is due to the fact that, even from a perceptive point of view,

[2] http://www.idiap.ch/HeadPoseDatabase/

discriminating between head tilts is more difficult than discriminating between head pan or head roll [2]. With respect to other works, these results are good. For instance, in [17], a neural net is used to train a head pose classifier from data recorded directly in two meeting rooms. When using 15 people for training and 2 for testing, average errors of 10 degrees in pan and tilt are reported. However, when training the models in one room and testing on data from the other meeting room, the average errors rise to 20 degrees. This suggests that their appearance model is fitted to their set-up, to the contrary of our experiments, in which appearance models are trained from an external database [5].

4 Visual Focus of Attention Modeling

VFOA set: For these experiments, we considered only the two persons which have their head pose continuously annotated using FOB (left person and right person). For each one of these person, we identified 6 dominant VFOA of interest and defined the set of visual VFOA as $\mathcal{F} = \{f_1 = person,\ f_2 = organizer1,\ f_3 = organizer2,\ f_4 = table,\ f_5 = slidescreen,\ f_6 = unfocused\}$. For left person f_1 is person right and for right person, f_1 is person left. In the following we present our VFOA modeling approaches.

4.1 Modeling VFOA with a Gaussian Mixture Model (GMM)

Let us denote by $F_t \in \mathcal{F}$ and by Z_t the VFOA and the head pointing vector (defined by its pan and tilt angles) of a person at time instant t. Estimating the VFOA can be posed in a probabilistic framework as finding the label maximizing the a posteriori (MAP) probability:

$$\hat{F}_t = \arg\max_{F_t \in \mathcal{F}} p(F_t|Z_t) \text{ with } p(F_t|Z_t) = \frac{p(Z_t|F_t)p(F_t)}{p(Z_t)} \propto p(Z_t|F_t)p(F_t) \quad (3)$$

For each possible VFOA $f \in \mathcal{F}$ which is not $unfocused$, $p(Z_t|F_t)$ is modeled as a Gaussian distribution $\mathcal{N}(Z_t; \mu_f, \Sigma_f)$ with mean μ_f and full covariance matrix Σ_f. Besides, $p(Z_t|F_t = unfocused)$ is modeled as a uniform distribution. For the distribution over priors, two alternatives were considered. In the first case, no prior was used (i.e. the distribution was uniform), while in the second case, the priors were learned from the considered training data.

4.2 Modeling VFOA with a Hidden Markov Model (HMM)

Modeling the VFOA with a GMM does not account for the temporal dependencies between the VFOA events. As a model of these dependencies, we considered the classical graphical model shown in Figure 2. Given a sequence of VFOA $F_{0:T} = \{F_t, t = 0, ..., T\}$ and a sequence of observations $Z_{1:T}$, the joint posterior probability density function of the states and observation can be written:

$$p(F_{0:T}, Z_{1:T}) = p(F_0) \prod_{t=1}^{T} p(Z_t|F_t)p(F_t|F_{t-1}) \quad (4)$$

The emission probabilities were modeled as in the previous case (i.e. Gaussian distributions for regular VFOA, and uniform distribution for the *unfocused* label). The parameters of these models, along with the discrete transition matrix $p(F_t|F_{t-1})$ modeling the probability to transit from a VFOA to another were learned from training data. In the testing phase, the estimation of the optimal sequence of states given a sequence of observations was conducted using Viterbi algorithm [14].

5 Experimental Results

5.1 Evaluation Set Up

To perform evaluate we annotated our IHPD database with the VFOA of person right and left by watching the videos. While there might be some issue about the feasibility of VFOA annotation based on eye gaze, experiments have shown that there might be more than 95% agreement among annotators for this task.

Evaluation protocol. For training and testing, we adopted the leave one out protocol. The data of 7 recordings were used to train the model parameters which are then used to test the recognition system on the remaining one. We defined two kinds of errors measure to evaluate the performances of our modeling.

Frame based recognition rate (FRR). Which corresponds to the percentage of correctly estimated VFOA frames and indicates the proportion of time when the VFOA has been correctly identified. This rate, however, can be dominated by long duration VFOA events. Since we are also interested in the VFOA events sequence patterns, which contains information related to the interaction, we also need a measure reflecting how well these events, short or long, are recognized.

Event based precision/recall, and F-measure. Let us consider two sequences of VFOA events, the GT sequence G obtained from the VFOA annotations and the recognized sequence R obtained through the VFOA estimation process. The GT sequence is defined as $G = \{G_i = (f_i, b_i, e_i),\ i = 1, ..., N_G\}$ where N_G is the number of events, $f_i \in \mathcal{F}$ is the ith VFOA event label, b_i and e_i the beginning and end time of the event f_i. The recognized sequence R is defined similarly. The two sequences are aligned using an adaptive string alignment procedure that take into account the temporal extent of the events. Given this alignment we can compute for each event $f \in \mathcal{F}$, the recall $\rho(f)$ and precision $\pi(f)$ measures of that event defined as:

$$\forall f \in \mathcal{F},\ \rho(f) = \frac{N_{matched}(f)}{N_G(f)} \text{ and } \pi(f) = \frac{N_{matched}(f)}{N_R(f)} \tag{5}$$

where $N_{matched}(f)$ represents the number of events in the recognized sequence labeled f that match the same event in the GT after alignment. $N_G(f)$ (resp $N_R(f)$) denotes the number of occurrences of the event f in the ground truth (resp recognized) sequence. The recall measures the percentage of ground truth events that are correctly estimated while the precision measure the percentage

Table 2. Average VFOA estimation results for right person using maximum likelihood estimation (ML), GMM modeling with and without prior term and HMM with prior (transition matrix learned from training data); gt=using ground truth, tr=using tracking output

error measure	gt-ML	gt-gmm	gt-gmm-prior	gt-hmm	tr-ML	tr-gmm	tr-gmm-prior	tr-hmm
frame rr	62.1	53.6	60.7	53.9	42.8	38.2	46.6	38.4
event rec	65.7	57.3	52.3	50.6	54.5	51.5	38.1	34.8
event prec	43.6	43.6	47.3	52.2	18.5	17.1	19.4	40.6
event F-meas	52.1	47.2	48.4	50.4	29.5	25.3	24.9	36.9

Table 3. Average VFOA estimation results for left person using maximum likelihood estimation (ML), GMM modeling with and without prior term and HMM with prior (transition matrix learned on training data); gt=using ground truth, tr=using tracking output

error measure	gt-ML	gt-gmm	gt-gmm-prior	gt-hmm	tr-ML	tr-gmm	tr-gmm-prior	tr-hmm
frame rr	78.4	73	75.3	73	53.6	49.5	51	50.1
event rec	66.9	62	55.5	56.4	51.3	39.3	39	32.7
event prec	53.2	56.8	53.1	63.8	26.8	18.9	20.2	44.9
event F-meas	59	58.7	53.8	59.2	34.2	25.2	25.5	36.9

of estimated events that are correct. Both precision and recall need to be high to denote good VFOA recognition performance. The F-measure defined by:

$$\phi(f) = \frac{2\rho(f)\pi(f)}{\rho(f) + \pi(f)} \tag{6}$$

reflects this requirement. Finally, the performance measures of a given person are computed through averaging: l

$$\rho = \frac{\sum_{f\in\mathcal{F}} \rho(f)}{|\mathcal{F}|},\ \pi = \frac{\sum_{f\in\mathcal{F}} \pi(f)}{|\mathcal{F}|},\ \text{ and } \phi = \frac{2\rho\pi}{\rho+\pi} \tag{7}$$

The performance measure over the whole database is the average of the precision, recall and F-measure of the 8 individuals.

5.2 Experimental Results

Results exploiting the head pose ground truth: In this section we provide the VFOA estimation results when the input data to the algorithms of Section 4 are the head poses obtained from the flock-of-birds sensors.

VFOA and head pose correlation: Table 2 and 3 display the VFOA estimation results for the right and left person respectively. The first column of theses two tables give the results of VFOA maximum likelihood estimation (ML) using the head pose GT data, where the ML approach consists in estimating the VFOA model parameters using the data of a person and testing the model on the same data (when considering the GMM modeling). These results show in an optimistic case the performances our model can achieve, and illustrate somehow the

correlation between a person's head poses and his VFOA. As can be seen, this correlation is quite high for the left person (close to 80% FRR), showing the good accordance between pose and VFOA. This correlation, however, drops to near 60% only for the right person. This can be explained by the fact that for person right, there is a strong ambiguity between looking at person left and at the slide screen (see Figure 1). More generaly, the range of azimuth values within which the three other participants and slide screen VFOA target lies has been divided by 2. The average angular distance between these targets is around 20 degrees for right person, a distance which can easily be compensated for using eye movements only (rather than head pose) to change focus. The values in the confusion matrices (not shown) corroborate this analysis. The analysis of Tables 2 and 3 shows that this discrepancy holds for all experimental conditions and algorithms (when using ground-truth input), with a performance decrease of approx. 20% and 10% for FRR and event F-measure respectively.

VFOA Prediction: While the MLE is achieving the best results, its performances are not extremely out-performing the performances of the GMM modeling with or without a prior term and the HMM modeling using GT. The GMM and HMM modeling are showing the ability to predict a person VFOA from other persons. For both person right and left, the GMM modeling is achieving better frame based recognition performances and event recall while the HMM is giving better event precision. This can be explained since the HMM approach is doing some data smoothing. As a results some events are missed (lower recall) but the precision increases due to the elimination of short spurious detections.

Influence of Priors: Figure 3 shows the effect of the prior on the VFOA distribution for person right. the VFOA $organizer2$, represented in this figure by the green area, have its surface reduced while the VFOA $personleft$ (black area), $organizer1$ (blue area), $slidescreen$ (yellow area), $table$ (red area) have their surface extended because these events are more likely. The prior forces the model to concentrate on most likely events while almost removing less likely events. The use of these priors could clearly be a problem when using the VFOA recognition system on other meetings.

Comparison to others: in the interesting work [12], the task, amongst others, consists of estimating the VFOA of four people engaged in conversation, using people's speaking status and head pose measured with magnetic sensors. For each person, the potential VFOA were the three other participants. They obtained an average frame based recognition rate of 67.9 %. Despite the lower number of target VFOA, their result is similar to ours (we obtained around 60% for person right and 75% for person left).

Results with Head Pose Estimates: Table 2 and 3 show also the results for VFOA estimation using tracking results, which exhibit performances degradation w.r.t. GT data. These degradation are mainly due to tracking errors (short periods when the tracker locks on a subpart of the face, tilt uncertainty) and the different head pose estimation tracker response to input with similar poses but different appearences. Figure 1 shows the effect of tilt estimation errors on the

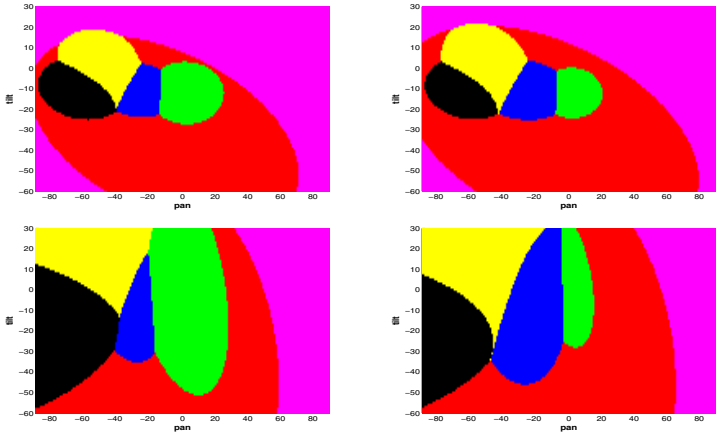


Fig. 3. Pan-tilt space VFOA maps using GT (first row) and tracking estimates (second row) for person right. First column: without prior; second column: with priors. black=$personleft$, yellow=$slidescreen$, blue=$organizer1$, green=$organizer2$, red=$table$, magenta =$unfocused$.

VFOA distributions. While the VFOA distributions in pan of the GT (first row) and the tracking estimates (second row) are similar, the VFOA distributions in tilt are wider for the tracking estimates. The tables also show that, while, when using GT head pose data, the HMM modeling did not have much impact on performances w.r.t. the GMM case, we observe from the reported event F-measure that in presence of noisier data, its smoothing effect is quite beneficial. The confusion matrices based on the tracking estimates exhibit the same trnds than when using the GT head pose, but with more pronounced confusion because of the tracking errors (see above) and tilt estimation uncertainties.

Our results -close de 50% frame recognition rate- is quite far from the 73% reported in [17]. Several factors may explain the difference. First, in [17], a 4 people meeting situation was considered and no other VFOA target appart from the other meeting participants was considered. Thus, the pan head angle was sufficient to differentiate VFOA targets. In addition, these participants were sitting at equally spaced angles around a round table, optmising the discrimination between VFOA targets. Finally, it seems from the paper that the head pose tracking algorithm was trained on the face images of the same people appearing in the test video, which resulted in smaller tracking errors.

6 Conclusion and Future Work

In this paper we presented a system to recognize the VFOA of meeting participants from their head pose, the latter being defined by its pan and tilt angles. Such head pose measurements were obtained using either through magnetic sensors or a probabilistic head pose tracking algorithm. The experiments showed

Fig. 4. Ambiguity in focus: despite the high visual similarity of the head pose of the right person, the two focus are different (left image: left person: right image: slide screen). Resolving such cases can only be done by using context (speaking status, other's people gaze, slide activity etc).

that, depending on people's position in the meeting room and on the angular distribution of the FOA targets, *the eye gaze may or may not be highly correlated with the head pose.* In absence of such correlation, the only way to improve VFOA recognition may come from the prior knowledge embedded in the cognitive and interactive aspects of human-to-human communication. Ambiguous situations such as the ones illustrated in Figure 4, where the same head pose can correspond to two different VFOA targets, could be resolved by the joint modeling of the speaking and VFOA characteristics of all meeting participants. which has been shown to exhibit specific patterns/statistics in cognitive studies. Besides, as there exists some correlation between head pose tracking errors and VFOA recognition results, we will conduct further research to improve the tracking algorithms, e.g. using multiple cameras or adaptive appearence modeling techniques. Finally, in the case of meetings in which people are moving to the slide screen or white board for presentations, the development of a more general approach that models the VFOA of these moving people will be necessary. This has been one topic of our recent research [16].

References

1. Ba, S.O., Odobez, J.-M.: A Rao-Blackwellized Mixed State Particle Filter for Head Pose Tracking. in Meetings. In Proc. of ACM ICMI Workshop on Multi-modal Multi-party Meeting Processing (MMMP), Trento Italy October 7, 2005
2. Brown, L., Tian, Y.: A study of Coarse Head Pose Estimation. In Proc. of IEEE Workshop on Motion and Video Computing,Orlando Florida, Dec 2002
3. Danninger, M., Vertegaal, R., Siewiorek, D., P., Mamuji, A.: Using Social geometry to manage interruptions and co-worker attention in office environments. In Proc. of Conference on Graphics Interface, Victoria, British Columbia, 2005
4. Duncan Jr., S.: some signals and rules for taking speaking turns in conversations. Journal of Personality and Social Psychology, 23(2),pp283-292, 1972
5. Gourier, N., Hall, D., CrowleyJ., L.: Estimating face orientation from robust detection of salient facial features. in Proc. of Pointing 2004, ICPR, International Workshop on Visual Observation of Deictic Gestures, Cambridge, UK

6. Heylen, D.: Challenges ahead head movements and other social acts in conversation. in Proc. of The Joint Symposium on Virtual Social Agent, 2005
7. Langton,S.R.H., Watt, R. J., Bruce, V.: Do the eyes have it? Cues to the direction of social attention. Trends in Cognitive Sciences, vol 4, no2, pp50-58, February 2000
8. Langton,S.R.H.: The mutual influence of gaze and head orientation in the analysis of social attention direction. Quarterly Jl of Exp. Psychology, 53A(3),825-845, 2000
9. Matsumoto, Y., Ogasawara, T., Zelinsky, A: Behavior recognition based on head pose and gaze direction measurement. In Conf. on Intel. Robots and Sys., 2002
10. MacGrath, J. E.: Groups: Interaction and performances Prentice-Hall, Inc., Englewoods Cliffs, N.J., 07632, 1984
11. Novick, D., Hansen, B., Ward, K.: Coordinating turn taking with gaze. In Proc. of International Conf. on Spoken Language Processing, october 1996
12. Otsuka, K., Takemae, Y., Yamato, J., Murase, H.: A probabilistic inference of multi party-conversation structure based on Markov switching models of gaze patterns, head direction and utterance. In Proc. of International Conf. On Multi-modal and Interfaces, Trento, 2005
13. Pieters, R. G. M., Rosbergen, E., Hartog., M.: Visual attention to advertising: The impact of motivation and repetition. In Proc. of Conf. on Advances in Consumer Research, 1995.
14. Rabiner, L. R.: A tutorial on hidden Markov models and selected applications in speech recognition. Readings in Speech Recognition, pp. 267–296, 1990
15. Smith,P., Shah, M., da Vitoria Lobo, N.: Determining driver visuual attention with one camera. IEEE Trans. on Intel. Transportation Systems, 4(4):205-218, Dec. 2004
16. Smith K., Ba S., Gatica-Perez D. and Odobez J.M.: Multi-Person Wandering-Focus-of-Attention Tracking. IDIAP Research Report 80, Nov., 2005.
17. Stiefelhagen, R., Yang, J., Waibel, A.: Modeling focus of attention for meeting indexing based on multiple cues. IEEE Trans. on Neural Net., Vol.13, No. 4, 2002.
18. Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I.: Modeling Individual and Group Actions in Meetings with Layered HMMs. IEEE Trans. on Multimedia, June, 2006.
19. Flock of Birds: http://www.ascension-tech.com/products/flockofbirds.php

Multi-person Tracking in Meetings: A Comparative Study

Kevin Smith[1], Sascha Schreiber[2], Igor Potúcek[3], Vítzslav Beran[3], Gerhard Rigoll[2], and Daniel Gatica-Perez[1]

[1] IDIAP Research Institute, Switzerland
[2] Technische Universität München, Germany
[3] Brno University of Technology, Czech Republic

Abstract. In this paper, we present the findings of the Augmented Multiparty Interaction (AMI) project investigation on the localization and tracking of 2D head positions in meetings. The focus of the study was to test and evaluate various multi-person tracking methods developed in the project using a standardized data set and evaluation methodology.

1 Introduction

One of the fundamental goals of the AMI project is to formally and consistently evaluate tracking methods developed by AMI members using a standardized data set and evaluation methodology. In a meeting room context, these tracking methods must be robust to real-world conditions such as variation in person appearance and pose, unrestricted motion, changing lighting conditions, and the presence of multiple self-occluding objects. In this paper, we present an evaluation methodology for gauging the effectiveness of various 2D multi-person head tracking methods and provide an evaluation of four tracking methods developed under the AMI framework in the context of a meeting room scenario.

The rest of this paper is organized as follows: Section 2 describes the method of evaluation, Section 3 briefly describes the tracking methods, Section 4 presents the results of the evaluation, and Section 5 provides some concluding remarks.

2 Evaluation Methodology

To objectively compare the tracking methods, a common data set was agreed upon (Sec. 2.1) and evaluation procedure [13] was adopted (Sec. 2.2).

2.1 Data Set

Testing was done using the AV16.7.ami corpus, which was specifically collected to evaluate localization and tracking algorithms[1]. The corpus consists of 16 sequences recorded from two camera angles in a meeting room using four actors.

[1] We are thankful to Bastien Crettol for his support with the collection, annotation, and distribution of the AV16.7ami corpus, and to the participants for their time.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 88–101, 2006.

Fig. 1. *Examples from seq14 of the AV16.7.avi data corpus.* Left: Typical meeting room data with four participants (free to stand, sit, walk). Center: Participant heads near the camera are not fully visible and often move in and out of the scene. Right: The data set also contained challenging situations such as this (four heads appear and are annotated in this image).

Seven sequences were designated as the training set, and nine sequences for testing. The sequences depict up to four people performing common meeting actions such as sitting down, discussing around a table, etc (see Figure 1). Participants acted according to different predefined agendas for each scene (they were told the order in which to enter the room, sit, or pass each other), but the behavior of the subjects was otherwise natural. The sequences contain many challenging phenomena for tracking methods including occlusion, cameras blocked by passing people, partial views of backs of heads, and large variations in head size (see Table 1).

The corpus was annotated using bounding boxes for head location for use in training and evaluation [3]. Annotators were instructed to fit the bounding boxes around the perimeters of the participants heads, which were ambiguous in some cases. To reduce annotation time, every 25th frame was annotated (evaluations were performed only on annotated frames).

2.2 Measures and Procedure

In [13], the task of evaluating tracker performance was broken into evaluating three tasks: fitting ground truth persons (or $\mathcal{GT}$s) with tight bounding boxes

Table 1. Challenges in the AV16.7.ami data corpus test set (yes = y, no = n)

	seq01		seq02		seq03		seq08		seq09		seq12		seq13		seq14		seq16	
	L	R	L	R	L	R	L	R	L	R	L	R	L	R	L	R	L	R
duration (sec)	63		48		208		99		70		103		94		118		89	
total # heads	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
frontal heads	1	1	1	1	1	1	2	0	2	0	3	0	3	0	2	2	4	2
rear heads	1	1	1	1	1	1	0	2	0	2	0	3	0	3	2	2	4	4
event: occlusion	n	n	n	n	n	n	y	n	y	y	y	y	y	y	y	y	y	n
event: camera blocked	y	y	y	y	n	n	y	y	n	y	n	y	n	y	y	y	y	y
event: sit down	n	n	n	n	y	y	y	y	n	n	y	y	y	y	y	y	n	n

(referred to as *spatial fitting*), predicting the correct number and placement of people in the scene (referred to as *configuration*), and checking the consistency with which each tracking result (or estimate, $\mathcal{E}$) assigns identities to a $\mathcal{GT}$ over its lifetime (referred to as *identification*). Several measures are defined to evaluate these tasks, each dependant on the fundamental *coverage test*. The tasks measured in [13] are similar in many ways to those in [7], but the methods for measuring differ in a fundamental way: the mapping of $\mathcal{E}$s and $\mathcal{GT}$s. Measures in [7] are computed using a one-to-one mapping, whereas [13] defines measures using many-to-one w.r.t. $\mathcal{E}$s and many-to-one w.r.t. $\mathcal{GT}$s. We believe the latter to be a superior method, since situations can arise where there is no clearly correct one-to-one mapping between the $\mathcal{E}$s and $\mathcal{GT}$s.

2.2.1 Coverage Test

The coverage test determines if a $\mathcal{GT}$ is being tracked by an $\mathcal{E}$, if a $\mathcal{E}$ is tracking a $\mathcal{GT}$, and reports the quality of the tracking result. For a given tracking estimate $\mathcal{E}_i$ and ground truth $\mathcal{GT}_j$, the coverage test measures the overlap between the two areas using the *fitting F-Measure* $F_{i,j}$ [11]

$$F_{i,j} = \frac{2\alpha_{i,j}\beta_{i,j}}{\alpha_{i,j} + \beta_{i,j}} \qquad \alpha_{i,j} = \frac{|\mathcal{E}_i \cap \mathcal{GT}_j|}{|\mathcal{GT}_j|} \qquad \beta_{i,j} = \frac{|\mathcal{E}_i \cap \mathcal{GT}_j|}{|\mathcal{E}_i|} \tag{1}$$

where recall (α) and precision (β), are well-known information retrieval measures. If the overlap passes a fixed coverage threshold ($F_{i,j} \geq t_c$, $t_c = 0.33$), then it is determined that $\mathcal{E}_i$ *is tracking* $\mathcal{GT}_j$ and $\mathcal{GT}_j$ *is tracked* by $\mathcal{E}_i$.

2.2.2 Configuration

In this context, configuration means the number, the location, and the size of all people in a frame. A tracking result is considered to be *correctly configured* if and only if exactly one $\mathcal{E}_i$ is tracking each $\mathcal{GT}_j$. Four types of errors may occur, which correspond to the four configuration measures:

- **FN** - False negative. A $\mathcal{GT}$ is which not tracked by an $\mathcal{E}$.
- **FP** - False positive. An $\mathcal{E}$ exists which is not tracking a $\mathcal{GT}$.
- **MT** - Multiple trackers. More than one $\mathcal{E}$ is tracking a single $\mathcal{GT}$. An MT error is assigned for each excess $\mathcal{E}$.
- **MO** - Multiple objects. An $\mathcal{E}$ is tracking multiple $\mathcal{GT}$s. An MO error is assigned for each excess $\mathcal{GT}$.

False negative (**FN**) False positive (**FP**) Multiple trackers (**MT**) Multiple objects (**MO**)

Fig. 2. The four types of configuration errors. $\mathcal{GT}$s are represented by green boxes, $\mathcal{E}$s by red and blue boxes.

An example of each error type is depicted in Fig. 2, where the $\mathcal{GT}$s are marked with green colored boxes, the $\mathcal{E}$s with red and blue. One can also measure the difference between the number of $\mathcal{GT}$s and the number of $\mathcal{E}$s:

– **CD** - Counting distance. For a given frame, the difference between the number of $\mathcal{E}$s ($N^t_{\mathcal{E}}$) and $\mathcal{GT}$s ($N^t_{\mathcal{GT}}$) normalized by the number of $\mathcal{GT}$s ($N^t_{\mathcal{GT}}$).

$$\mathbf{CD} = \frac{N^t_{\mathcal{E}} - N^t_{\mathcal{GT}}}{max(N^t_{\mathcal{GT}}, 1)} \tag{2}$$

2.2.3 Identification

In the context of this evaluation, identification implies the persistent tracking of a $\mathcal{GT}$ by a particular $\mathcal{E}$ over time. Though several methods to associate identities exist, we adopt an approach based on a *majority rule* [13]. A $\mathcal{GT}_j$ is said to be identified by the $\mathcal{E}_i$ which passes the coverage test for the majority of $\mathcal{GT}_j$s lifetime, and similarly $\mathcal{E}_i$ is said to identify the $\mathcal{GT}_j$ which passes the coverage test for the majority of $\mathcal{E}_i$s lifetime (this implies that associations between $\mathcal{GT}$s and $\mathcal{E}$s will not necessarily match).

There can arise two types of identification failures, quantified by five measures.

– **FIT** - Falsely identified tracker. Occurs when a $\mathcal{E}_k$ which passed the coverage test for $\mathcal{GT}_j$ is not the identifying tracker, $\mathcal{E}_i$. FITs often result when $\mathcal{E}_i$ suddenly stops tracking $\mathcal{GT}_j$ and another $\mathcal{E}_k$ continues tracking $\mathcal{GT}_j$.
– **FIO** - Falsely identified object. Occurs when a $\mathcal{GT}_k$ which passed the coverage test for $\mathcal{E}_i$ is not the identifying person, $\mathcal{GT}_j$. FIOs often result from swapping $\mathcal{GT}$s, i.e. $\mathcal{E}_i$ initially tracks $\mathcal{GT}_j$ and subsequently tracks $\mathcal{GT}_k$.
– **OP** - Object purity. If $\mathcal{GT}_j$ is identified by $\mathcal{E}_i$, then OP is the ratio of frames in which $\mathcal{GT}_j$ and $\mathcal{E}_i$ passed the coverage test ($n_{i,j}$) to the overall number of frames $\mathcal{GT}_j$ exists (n_j).
– **TP** - Tracker purity. If $\mathcal{E}_i$ identifies $\mathcal{GT}_j$, then TP is the ratio of frames in which $\mathcal{GT}_j$ and $\mathcal{E}_i$ passed the coverage test ($n_{j,i}$) to the overall number of frames $\mathcal{E}_i$ exists (n_i).
– *identity F-Measure* - combines **OP** and **TP** using the F-measure such that if either component is low, identity F-Measure is low: $identity\ FMeasure = \frac{2\ OP\ TP}{OP+TP}$.

2.2.4 Procedure

To evaluate the ability of each tracking method for the tasks of spatial fitting, configuration and identification over diverse data sets, the following procedure is followed for each sequence:

Evaluation procedure for a data sequence.

1. `for` each frame in the sequence
   – `determine` *tracking maps* by applying the coverage test over all combinations of $\mathcal{E}$s and $\mathcal{GT}$s.

   - `record` configuration measures (FN,FP,MT,MO, CD) and fitting F-Measure from tracking maps.
2. `determine` *identity maps* for tracked $\mathcal{E}$s and $\mathcal{GT}$s using the *majority rule*.
3. `for` each frame in the sequence
   - `record` identification errors (FIT,FIO) from the identity maps.
4. `normalize` the configuration and identification errors and `compute` the purity measures for the entire sequence (the instantaneous number of ground truths and estimates are $N_{\mathcal{GT}}$ and $N_{\mathcal{E}}$ respectively, and the total number of frames is T).

$$\overline{FP} = \frac{1}{T}\sum_{t=1}^{T}\frac{FP_t}{max(N_{\mathcal{GT}}^t,1)}\ ,\ \overline{FN} = \frac{1}{T}\sum_{t=1}^{T}\frac{FN_t}{max(N_{\mathcal{GT}}^t,1)},$$

$$\overline{MT} = \frac{1}{T}\sum_{t=1}^{T}\frac{MT_t}{max(N_{\mathcal{GT}}^t,1)}\ ,\ \overline{MO} = \frac{1}{T}\sum_{t=1}^{T}\frac{MO_t}{max(N_{\mathcal{GT}}^t,1)},$$

$$\overline{FIT} = \frac{1}{T}\sum_{t=1}^{T}\frac{FIT_t}{max(N_{\mathcal{GT}}^t,1)}\ ,\ \overline{FIO} = \frac{1}{T}\sum_{t=1}^{T}\frac{FIO_t}{max(N_{\mathcal{GT}}^t,1)},$$

$$\overline{OP} = \frac{1}{N_{\mathcal{GT}}}\sum_{j=1}^{N_{\mathcal{GT}}}\frac{n_{i,j}}{n_j}, \qquad \overline{TP} = \frac{1}{N_{\mathcal{E}}}\sum_{i=1}^{N_{\mathcal{E}}}\frac{n_{j,i}}{n_i}, \qquad \overline{CD} = \frac{1}{T}\sum_{t=1}^{T}|\mathbf{CD}|$$

Note that most measures are normalized by $N_{\mathcal{GT}}$ and the number of frames (such as $\overline{FP}$). For these measures, the number reported could be thought of as a rate of error. For instance, $\overline{FP} = .25$ could be interpreted as: "for a given person, at time t, 0.25 FP errors will be generated on average."

3 Tracking Methods

Four head tracking methods built within AMI were applied to the data corpus and evaluated as described in Section 2. Each method approached the problem of head tracking differently, and it is noteworthy to list some of the qualitative differences (see Table 2). These methods are described briefly below.

3.1 Method A: Trans-Dimensional MCMC (developed at IDIAP)

Method A uses an approach based on a hybrid Dynamic Bayesian Network that simultaneously infers the number of people in the scene and their locations [12]. The state contains a varying number of interacting person models, each consisting of a head and body model. The person models evolve according to a dynamical model and a Markov Random Field (MRF) based interaction model (to prevent trackers from overlapping). The observation model consists of a set of global binary and color observations as well as individual head silhouette observations (to localize heads). The function of the global binary observation model is to predict the number of people in the scene. Inference is done by trans-dimensional Markov Chain Monte Carlo (MCMC) sampling (because of its ability to add/remove people from the scene and its efficiency).

3.2 Method B: Probabilistic Active Shape (developed at TUM)

Method B uses a double-layered particle filtering (PF) technique [5,6] consisting of a control layer (responsible for the detection of new people and evaluating the person configuration) and a basic layer (responsible for building a local probability distribution for each head). Locations for new people are derived from skin colored regions, which are detected using a normalized rg skin color model. Heads are modeled using a deformable active shape model consisting of 20 landmark points [1,2]. The basic layer PF samples and predicts a set of hypotheses for each person. Using the active shape model, a likelihood for the existence of a head in the image represented by the respective hypothesis can be computed. These sets of hypotheses are passed to the control layer PF, which evaluates and determines the configuration of heads by incorporating skin color validation and the local likelihood to verify the number of people being tracked.

3.3 Method C: KLT (developed at TUT)

Method C, proposed in [4] is based on the KLT feature tracker [8]. The method works by searching for potential people through performing background subtraction and skin color detection (using an RG skin color model) on the raw image. Connected component analysis is performed on the segmented image to find patches suitable for head detection. Ellipse-like shapes are then fitted to the patches and define a set of head centers. A KLT tracker, which extracts meaningful image features at multiple resolutions and tracks them by using a Newton-Raphson minimization method to find the most likely position of image features in the next frame, is initialized at each head center. Additionally, a color cue and rules for flocking behavior (alignment, separation, cohesion, and avoidance) are used to refine the tracking.

3.4 Method D: Face Detector (developed at BUT)

Method D, proposed in [10], is based on skin color segmentation and face detection. A learned skin color model is used to segment the image. Connected component analysis and morphological operations on the skin color segmented

Table 2. Properties of the various head tracking approaches

	Method A	Method B	Method C	Method D
Learned Models	binary, color, head shape	skin color, shape	skin color	face/nonface weak classifiers
Initialization	automatic	automatic	automatic	automatic
Features	background sub, silhouette, color	motion detection, skin color, head/shoulder shape	background sub, skin color, local charact.	skin color, gabor wavelets
Mild Occ.	robust	robust	robust	robust
Severe Occ.	semi-robust	semi-robust	sensitive	sensitive
Identity Recovery	swap, rebirth	swap, rebirth	rebirth	none
Comp. Exp.	∼1 frame/sec	∼3 frame/sec	∼20 frame/sec	∼0.2 frame/sec

image are used to propose head locations. Face detection is then applied to the skin color blobs to determine the likelihood of the presence of a face. The face detection is based on the well-known AdaBoost [14] algorithm which uses weak classifiers to classify an image patch as a face or non-face. Method D replaces the simple rectangular image features with more complex Gabor wavelets [9]. The face detector was trained on normalized faces from the CBCL data set (1500 face and 14000 non-face images) and outputs a confidence, which is then thresholded to determine if a face exists. Faces are associated between frames using a proximity association defined on the positions of the detected faces.

4 Evaluation

The four methods were evaluated for their performance at the tasks outlined in Section 2: spatial fitting, configuration, and identification. Methods A and B were tested on 360×288 non-interlaced images; Methods C and D were tested on 720×576 interlaced images after applying an interpolating filter. This discrepancy may affect the relative performance of the methods, but we believe the effect to be minimal. In the following, we present a summary of the overall performance of the tracking methods, followed by a detailed discussion of each task[2].

4.1 Overall Performance

The fitting F-Measure is an indicator of the spatial fitting (see Figure 3). Spatial fitting refers to how tightly the $\mathcal{E}$ bounding boxes fit the $\mathcal{GT}$. The fitting F-Measure is only computed on correctly tracked people, and a value of one indicates perfectly fit bounding boxes. Lower numbers indicate looser, misaligned, or missized tracking estimates. Results for the fitting F-Measure indicate that methods A and D performed comparably well at about .60. Measures B and C performed at approximately .50. The spatial fitting depends on many aspects of the method including the features, motion model, and method of inference. Intuition suggests that the boosted Gabor wavelets of Method D and the head silhouette feature of Method A were most precise in this case, but these results cannot be solely attributed to these features without further experiments.

The counting distance $\overline{CD}$ measures the difference between the number of $\mathcal{GT}$s and $\mathcal{E}$s for a given frame, and gives an imperfect estimation of the configuration performance, i.e. the ability of the method to place the correct number of $\mathcal{E}$s in the correct locations. $\overline{CD}$ is an imperfect summary because some types of errors such as FPs and FNs may cancel in the calculation of $\overline{CD}$, but it is still a good indicator. The quantity $1 - \overline{CD}$ is reported so that higher numbers indicate better configuration performance ($\overline{CD} \in [0, \infty)$ but in our experiments ranged from 0 to 1). Methods A and C performed best, at about .73, while method D performed at .64, and B at .58. An alternative way to measure the overall configuration performance is to sort the methods by rankings of the individual

[2] Example videos and details can be found at http://www.idiap.ch/∼smith/

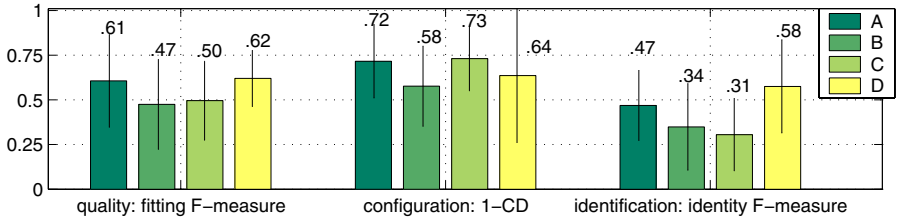


Fig. 3. Results for the three tracking tasks (spatial fitting, configuration, and identification). The *fitting F-measure* shows the spatial fitting, or tightness of the bounding boxes. The quantity $1 - \overline{CD}$ is indicative of the ability of a method to estimate the configuration. The ability of a method to maintain consistent identities is measured by the *identity F-Measure*. The numbers above each bar represent the mean for the entire data set, and the lines represent the standard deviations.

configuration measures (see Section 4.3 and Figure 5). Doing so, we find that Method C performs the best, followed by Method A, Method D, and finally Method B. Though not necessarily so, in this case this result is consistent with the findings of the counting distance.

The *identity F-Measure* measure indicates how consistently a method was able to identify the $\mathcal{GT}$s over time; it is a combination of the $\overline{TP}$ and $\overline{OP}$ measures. In this case, method D clearly outperformed the others. This is somehow surprising, as it uses the simplest procedure for maintaining identity (spatial proximity between frames). More sophisticated methods such as models for swapping identities in Methods A and B, are perhaps not suited for this data. One the other hand, because Method D relies on specialized face detection, it's superior performance may not generalize to situations in which faces are not the target objects.

4.2 Spatial Fitting

As mentioned in Section 4.1, the fitting F-measure indicates the tightness of the fit of the bounding boxes to the $\mathcal{GT}$s. From Figure 4, it is apparent that certain sequences presented much more of a challenge than others. Figure 4 illustrates the variation of performance on specific pieces of data, something hidden by all-inclusive measures. Typically, fitting F-Measure values were similar for all the trackers at approximately 0.80, but for more challenging sequences such as 08R, 09R, 12R, 13R, and 16R, differences were more pronounced and fitting F-Measure values dipped as low as 0 in one case. Method D was the most spatially robust for the challenging sequences.

4.3 Configuration

Results for the four configuration error types and $\overline{CD}$ can be found in Figure 5.

The measure $\overline{FN}$ gives an estimation of the number of False Negatives (or undetected person ground truths) per ground truth, per frame. Method C performed the best in this respect, with .26 FN's per person, per frame. This

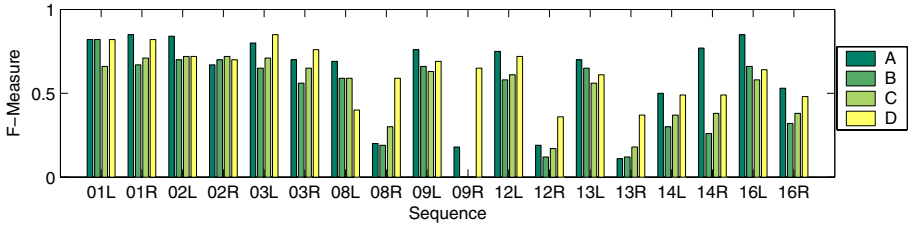


Fig. 4. The fitting F-Measure shows how tightly the estimated bounding boxes fit the ground truth (when passing the coverage test)

low rate of missed $\mathcal{GT}$s may be attributed to KLT trackers selection of meaningful image features. Method B performed significantly worse, averaging approximately .49 FN, which may be due to difficulties in fitting the contour to the appearance of some heads. FNs were the most prominent type of configuration error among all four tracking methods, usually as a result of an unexpected change in the appearance of a head, partial views, lighting changes, entrances/exits, and size variations and occlusions (sometimes as extreme as in Figure 1).

The measure $\overline{FP}$ estimates the number of False Positive errors (or extraneous $\mathcal{E}$s) per ground truth, per frame. This was the second most common type of configuration error. Typical causes for FP errors include face-like or skin colored objects in the background (texture or color), shadows, and background motion. Methods A and B were least prone to FP errors, with a rate of 0.08 FPs per person, per frame. Method A's low rate of FP errors can be attributed to the use of a body model, which only adds people when a body is detected (bodies are easier to detect than heads). This was followed by Method C with 0.21, and Method D with 0.23. Method D was particularly sensitive to FP generating conditions, as the standard deviation was roughly twice the mean, 0.42. Method D's FPs were generated by face-like or skin colored objects in the background and exposed skin on the arms of the participants.

The measure $\overline{MT}$ estimates the number of Multiple Tracker errors (which occur when several estimates are tracking the same ground truth person). The only method significantly prone to this type of error was Method A. This susceptibility is due to the fact that Method A uses strong priors on the size of the body and head to help the foreground segmented image features localize the head. The priors of Method A are trained using participants in the far field of view, and are not robust to dramatic changes in size. When a participant appears close to the camera, Method A often fits multiple trackers to the larger head area. Methods B,C, and D do no suffer from this effect because they do not enforce constraints on the size of the head so strongly.

The measure $\overline{MO}$ estimates the number of Multiple object errors (which occur when one estimate tracks several ground truths) per person, per frame. This type of error generally occurs when a tracker estimate is oversized and expands to cover large areas of the image, or occasionally when people are near one another.

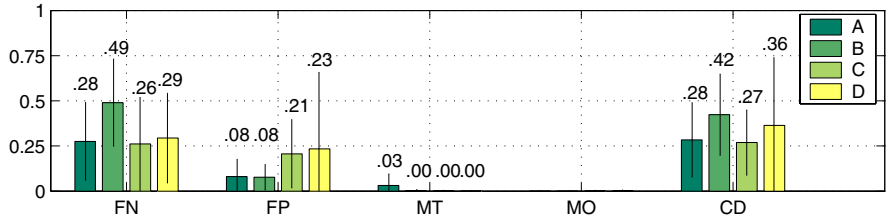


Fig. 5. The configuration measures, $\overline{FN}$, $\overline{FP}$, $\overline{MT}$, $\overline{MO}$, and $\overline{CD}$, normalized over the test set

All four methods tested were robust to this type of error. This robustness can be attributed to the modeling of head objects, interaction models, and motion models built into each of the methods.

The counting distance measure $\overline{CD}$ is described in Section 4.1.

4.4 Identification

Results for the identification measures can be found in Figure 6.

The $\overline{FIO}$ measure estimates the rate of Falsely Identified Object errors (when an $\mathcal{E}$ tracks a $\mathcal{GT}_k$ which is not the $\mathcal{GT}_j$ that the $\mathcal{E}$ *identifies*). Of the two types of identification errors (FIO and FIT), FIO errors occurred less frequently for all four methods. FIO errors are often generated when an $\mathcal{E}$ outlives the $\mathcal{GT}$ it is supposed to identify, and the $\mathcal{E}$ begins to track another $\mathcal{GT}$, though this was rare in our experiments. The other common mode of failure occurred when $\mathcal{E}$s confused $\mathcal{GT}$s, often as a result of occlusion. This method of failure was seen most in Methods A and B with $\overline{FIO}$ rates of 0.05 and 0.04, respectively. Interestingly, both these methods modeled *identity swapping*, where $\mathcal{E}$s switch labels in an attempt to maintain identity. Spurious identity swaps could account for higher FIO rates. Method C was very robust to FIO errors, with a negligible FIO rate. Method D was nearly as robust, with a $\overline{FIO}$ of 0.01.

The $\overline{FIT}$ measure reports the rate of Falsely Identified Tracker errors (which occur when a $\mathcal{GT}$ person is being tracked by a non-identifying $\mathcal{E}$). There are two typical sources of FIT errors. The first occurs, as with the FIO error, when $\mathcal{E}$s

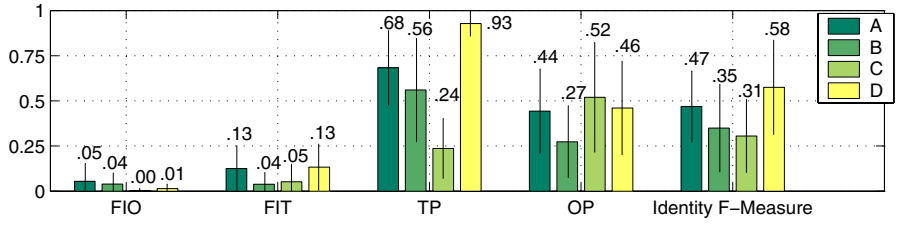


Fig. 6. The identification measures, $\overline{FIO}$, $\overline{FIT}$, $\overline{TP}$, $\overline{OP}$, and *identity F-Measure* computed over the test set

swap or confuse $\mathcal{GT}$s. The second error source occurs when several short-lived $\mathcal{E}$s track the same $\mathcal{GT}$s. Both of these sources caused FIT errors in our test set, though it can be expected that FIT contributions from the first error source should roughly match the FIO error rate (and thus, any increase in the FIT over the FIO is caused by short-lived $\mathcal{E}$s). Methods A and D saw the most FIT errors, with $\overline{FIT}$ rates at 0.13 (0.13 FIT errors are generated per frame, per person). Method D's FIT errors can be almost exclusively attributed to multiple, short-lived $\mathcal{E}$s tracking the same $\mathcal{GT}$. Method B was the most robust to FIT errors with a rate of 0.04.

The $\overline{TP}$ measure evaluates the consistency with which an $\mathcal{E}$ identifies a particular $\mathcal{GT}$. Mis-identified $\mathcal{GT}$s cause FIO errors, but the TP measure gives equal weight to all tracking estimates. $\mathcal{E}$s with a short

lifetime will not significantly influence the $\overline{FIO}$, and $\mathcal{E}$s with long lifetimes will dominate. Typically, in our experiments, the methods reported a higher $\overline{TP}$ than $\overline{OP}$. This indicates more $\mathcal{E}$s were generated than the number of $\mathcal{GT}$s in the sequence (in a temporal sense), and that they lasted for shorter lifetimes. Method D reported a $\overline{TP}$ of 0.93, which indicates that nearly all its $\mathcal{E}$s perfectly identified their $\mathcal{GT}$s. However, this does not indicate near-perfect identification. Method D's $\overline{OP}$, 0.46, while on par with the other methods, indicates that the $\mathcal{GT}$s were often tracked by multiple short-lived $\mathcal{E}$s. Method A reported the next highest $\overline{TP}$, with a value of 0.68, followed by Method B (0.56) and Method C (0.24). Method C was the only method to report a lower $\overline{TP}$ than $\overline{OP}$.

The $\overline{OP}$ measure evaluates the consistency with which a $\mathcal{GT}$ is identified by the same $\mathcal{E}$. Mis-identifying $\mathcal{E}$s can cause FIT errors, but OP gives equal weight to all $\mathcal{GT}$s in the sequence. Short-lived $\mathcal{GT}$s will not significantly affect the $\overline{FIT}$, and $\mathcal{GT}$s with a long lifetime will dominate. Method C reported the best $\overline{OP}$.

4.5 Summary and Qualitative Comments

Giving equal weight to the three tracking tasks described in this document (configuration, identification, and spatial fitting) and using a simple ranking system, the best performing tracking method is D, followed by A, C, and B. Method D is the most reliable at identification and exhibits the highest spatial fitting. However, it does have several drawbacks. It is the slowest of the four methods and the most sensitive to occlusion. The face detector is based on skin color detection and is more sensitive to lighting conditions than the other methods. Skin-colored segments of the background pose a problem for the face detector (Method D exhibits the highest $\overline{FP}$), and the $\overline{FN}$ suffers as the detector struggles with non-frontal faces.

Ranked second among the four methods is Method A. Method A was the only method which did not model skin color, and was the only method which modeled the body to help localize the head. The use of a body model had several effects. First, Method A had the lowest $\overline{FP}$ rate, which can be attributed to the body

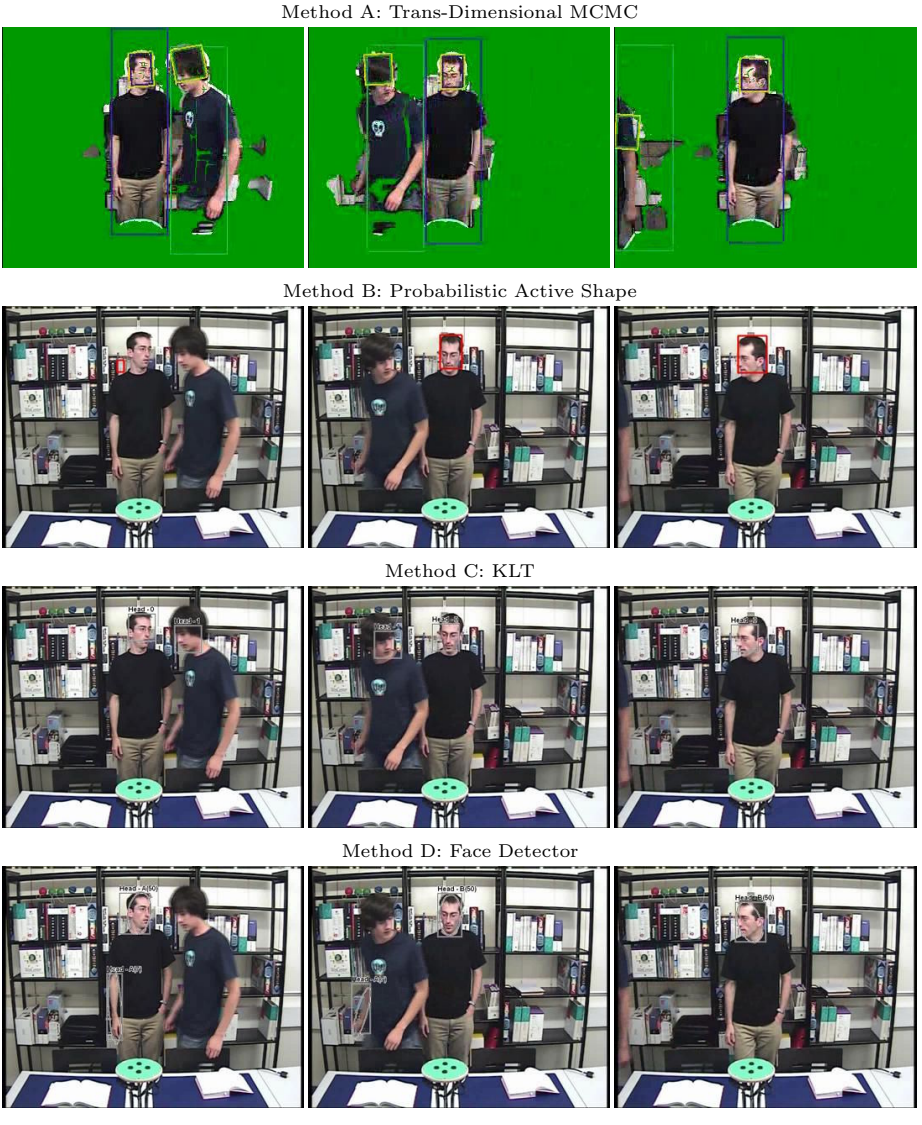


Fig. 7. Results for frames 307, 333, and 357 of sequence 09L from the AV16.7.avi data corpus. Method A: body and head results shown. A FP error appears in frame 357. Method B: heads results appear as red bounding boxes. Two FN errors and an FP error occur in 307, and one FN error occurs in 333. Method C: head results appear as grey bounding boxes. Method D: results appear as grey bounding boxes, participant arms are mistaken for heads in 307 and 333.

model preventing spurious head $\mathcal{E}$s. The body model assisted in detecting heads, which kept the $\overline{FN}$ rate low. However, because of strong size priors on the head and body models, Method A performed poorly when tracking heads near the

camera (resulting in MT errors). Method A was ranked second in spatial fitting and was also ranked second in maintaining identity, though incorrect swapping of $\mathcal{E}$ labels may have lowered this performance.

Method C was third overall among the four methods. It was the fastest computationally; the only one approaching real-time frame rates. Method C had the highest configuration performance, boasting the lowest FN rate and negligible MT and MO errors. This can be attributed to the KLTs selection of meaningful image features. However, Method C performed worst in terms of spatial fitting and identification. The poor spatial fitting might be due to a lack of shape features or features specialized to the face (as in the face detector). Problems with identification were due to the lack of an explicit way to manage identity among the trackers.

Finally, Method B fell last overall, but ranked third for each of the three tracking tasks. In terms of spatial fitting, Method B was the highest performing method for several of the sequences, but suffered from poor performance on some of the more difficult multi-person sequences (12R, 14R, and 16R). Among the four trackers, the Method B was the most robust to partial occlusions. For Method B, identity was maintained by binning gray values of the face shape. A lack of color information, poor shape adjustment, and a swapping mechanism like that of Method A, may have caused identification problems for this method.

From this evaluation, we might draw some of the following conclusions:

1. Shape-based methods, such as B and C, perform as well or better at spatial fitting when stable, but are more prone to configuration failures, and less able to recover from such failures.
2. Methods employing background subtraction (such as A and C) seem to have an advantage estimating the configuration of the scene.
3. Attempts to model identity changes to handle difficult tracking scenarios such as dramatic changes in size and appearance or frequent occlusions may do more harm than good (as for Methods A and B).

5 Conclusion and Future Work

The AV16.7.ami corpus contains many difficult real-life scenarios which remain challenging for state-of-the-art tracking methods. These results represent the first evaluation of methods for multi-person tracking in meetings using a common data set in the context of the AMI project. Future work might incorporate multi-model information or concentrate on tracking other objects in different scenarios.

Acknowledgements. This work was supported by the Swiss National Center of Competence in Research on Interactive Multimodal Information Management (IM2), and the EC project Augmented Multi-party Interaction (AMI, publication AMI-175).

References

1. T. Cootes and C. Taylor, *Statistical models of appearance for computer vision*, 2004.
2. T. Cootes, G. Edwards and C. Taylor, "A comparative evaluation of active appearance model algorithms", *British Machine Vision Conference*, Southampton, UK, Sept. 1998.
3. D. Gatica-Perez "Annotation Procedure for WP4-locate", *AMI Internal Document*, Martigny, Switzerland, October 2004.
4. M. Hradis, R. Juranek, "Real-time Tracking of Participants in Meeting Video", *Proceedings of CESCG*, Wien, 2006.
5. M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking", *International Journal of Computer Vision 29(1)*, pp. 5–28, 1998.
6. M. Isard and A. Blake, "A Mixed-State CONDENSATION Tracker with Automatic Model-Switching", *International Conference on Computer Vision (ICCV)*, 1998.
7. R. Kasture et. al., "Performance Evaluation Protocol for Face, Person, and Vehicle Detection & Tracking Analysis and Content Extraction (VACE-II)", ARDA Technical Report, Tampa, FL, 2006.
8. M. Kölsch and M. Turk, "Fast 2D Hand Tracking With Flocks and Multi Cue Integration", Department of Computer Science, University of California, 2005.
9. V. Kruger, "Wavelet Networks for Object Representation," thesis dissertation, Technischen Fakultat, Christian-Albrechts-Universitat zu Kiel, 2000.
10. I. Potucek, S. Sumec, M. Spanel, "Participant activity detection by hands and face movement tracking in the meeting room", *Computer Graphics International (CGI)*, Los Alamitos, 2004.
11. C.J. Van Rijsbergen, *Information Retrieval*, Butterworth-Heinemann, Newton, MA, USA, 1979.
12. K. Smith, S. Ba, J.M. Odobez, D. Gatica-Perez, "Multi-Person Wander-Visual-Focus-of-Attention Tracking", *IDIAP-RR-05-80*, Nov 2005.
13. K. Smith, S. Ba, J.M. Odobez, D. Gatica-Perez, "Evaluating Multi-Object Tracking", *CVPR Workshop on Empirical Evaluation Methods in Computer Vision (EEMCV)*, San Diego, CA, June 2005.
14. J. Viola and M. Jones, "Robust Real-time Object Detection", Technical Report 2001/01, Compaq CRL, February 2001.

Gaussian Mixture Models for CHASM Signature Verification

Andreas Humm, Jean Hennebert, and Rolf Ingold

Université de Fribourg, Boulevard de Pérolles 90, 1700 Fribourg, Switzerland
andreas.humm,jean.hennebert,rolf.ingold@unifr.ch

Abstract. In this paper we report on first experimental results of a novel multimodal user authentication system based on a combined acquisition of online handwritten signature and speech modalities. In our project, the so-called CHASM signatures are recorded by asking the user to utter what he is writing. CHASM actually stands for Combined Handwriting and Speech Modalities where the pen and voice signals are simultaneously recorded. We have built a baseline CHASM signature verification system for which we have conducted a complete experimental evaluation. This baseline system is composed of two Gaussian Mixture Models sub-systems that model independently the pen and voice signal. A simple fusion of both sub-systems is performed at the score level. The evaluation of the verification system is conducted on CHASM signatures taken from the MyIDea multimodal database, accordingly to the protocols provided with the database. This allows us to draw our first conclusions in regards to time variability impact, to skilled versus unskilled forgeries attacks and to some training parameters. Results are also reported for the two sub-systems evaluated separately and for the global system.

1 Introduction

Multimodal biometrics has raised a growing interest in the industrial and scientific community. The potential increase of accuracy combined with better robustness against forgeries makes indeed multimodal biometrics a promising field. In our work, we are interested in building multimodal authentication systems using speech and signatures as modalities. Speech and signatures are indeed two major modalities used by humans in their daily transactions and interactions. On the one hand, handwritten signatures are nowadays legally and socially accepted means for user authentication and contractual terms acceptation. On the other hand, producing a speech signal is a very natural non-intrusive gesture.

1.1 Motivations

Many automated biometric systems based on signature or speech alone have been studied and developed [9] [16]. However, we still see few deployments in commercial applications. We have identified three major reasons for this:

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 102–113, 2006.

(1) negative impact of time-variability [11], (2) degraded performances in the case of trained forgeries [10] [18], (3) decreased performances in mismatched conditions of use, such as mismatched sensors or mismatches environments [18]. While points (2) and (3) can be somehow handled with a minimum of supervision and control of the acquisition environment, the first point mentioned above has a critical impact for institutions willing to deploy such biometrics. Indeed, repeated enrollment sessions are not at all convenient for the user and generate further costs as they need to be secured.

We propose here a new approach to circumvent these problems while keeping an acceptable solution for the end user. The proposal is to record bimodal signatures by asking the user to simultaneously say and write the signature. Such bimodal signatures have already been presented in our preliminary work *Combined Handwriting and Speech Modalities for User Authentication* [6] and are referred here as *CHASM signatures*. In a similar way, we have also defined *CHASM handwriting* where the user reads what he is writing. CHASM handwriting could be used for user authentication or for enhanced content recognition, but this is out of the scope of this paper where we focus on CHASM signatures.

The main motivation of CHASM is therefore to increase performance and robustness by using two modalities instead of one. This work and our future work will attempt to assess in which degree CHASM signatures authentication systems can reach this goal. The motivation of performing a synchronized acquisition is multiple. Firstly, it avoids doubling the acquisition time. Secondly, the synchronized acquisition will probably give better robustness against intentional imposture as imitating simultaneously the voice and the writing of somebody else has a larger cognitive load. Finally, the synchronization patterns (i.e. where do users synchronize) or the intrinsic deformation of the inputs (mainly the slowdown of the speech) may be dependent to the user, therefore bringing useful biometrics information.

1.2 Related Work

Several related works have already shown that using speech and signature modalities together permits to improve significantly the authentication performances in comparison to systems based on speech or signature alone.

In [8], a tablet PC system based on online signature and voice modalities is proposed to ensure the security of electronic medical records. Tablet PCs are already used by many health care professional to have a patient's record readily available when prescribing or administering treatment. In this system, the user claims his identity by saying his first and last name that are recognized using speech recognition. The same waveform is then used with a speaker verification system based on GMMs to produce a score. In this way, the identification and verification steps are performed simultaneously. A signature is then acquired and a dynamic time warping verification system is used to produce a score. Speech and signature scores are then normalized and fused.

In [3], an online signature verification system and a speaker verification system are also combined. Both sub-systems use Hidden Markov Models (HMMs)

to produce independent scores that are then fused together. Results are reported for the two sub-systems evaluated separately and for the global system. Better accuracy is reported for the fused bimodal system. For this test, fictitious users are built by randomly associating signature and speech samples from two independent databases, namely *Philips' online signature database* and *Polyphone* and *Polyvar*.

In [11], tests are reported for a system where the signature verification part is built using HMMs and the speaker verification part uses either dynamic time warping or GMMs. The fusion of both systems is performed at the score level and results are again better than for the individual systems. This last work uses the BIOMET database [4] where the speech and signature data are recorded from the same user.

The main difference between these works and our CHASM approach lies in the acquisition procedure. In our case, the speech and signature data streams are recorded simultaneously, asking the user to actually say the content of his signature. Our procedure has the advantage to shorten the enrollment and access time and will potentially allow for more robust fusion strategies upstream in the processing chain.

The remainder of this paper is organized as follows. In section 2, we give an overview of the CHASM signature database used in this work and of the evaluation protocols. In section 3 we introduce GMMs and how they are used to model the speech and signature data streams. Section 4 presents the experimental results of the evaluation of our CHASM signature verification system. Finally, conclusions, discussions and future work are presented.

2 CHASM Signature Database

In this section we describe the database that we used to conduct the evaluation. Some comments on CHASM signature data are given and the evaluation protocols are then described.

2.1 MyIDea Database

CHASM data have been acquired in the framework of the MyIDea biometric data collection [2] [5]. MyIDea database is a multimodal database that contains other modalities such as fingerprint, talking face, palm print, etc. MyIDea contains about 70 users that have been recorded over three sessions spaced in time. In MyIDea, CHASM data have been recorded according to two scenarios. In the first one, a bimodal signature with voice is acquired. In this case, the user is actually asked to say the content of his signature, - *CHASM signature*. In the second scenario, the user is asked to write and read synchronously the content of a text, - *CHASM handwriting*. The data set used to perform the experiments reported in this article has been given the reference MYIDEA-CHASM-SET1 by the distributors of MyIDea. This set should be considered as a development set. A second set of CHASM data is planned for acquisition in a near future and will be used as evaluation set.

In MyIDea, CHASM data have been acquired with a WACOM Intuos2 graphical tablet and a standard computer headset microphone (Creative HS-300). For the signature stream, x,y-coordinates, pressure and the azimuth and elevation angles of the pen are sampled at 100 Hz. The speech waveform is recorded at 16 kHz and coded linearly on 16 bits. The data samples are also provided with timestamps to allow a precise synchronization of both streams. The timestamps are especially important for the signature streams as the graphical tablet does not send data samples when the pen is out of range. Fig. 1 shows an example of CHASM signature. The grey areas on the figure correspond to inter-stroke moments, when the user lift the pen out of the range of the tablet. We have to note that these kind of events are not very frequent for signatures and are more frequent for handwriting.

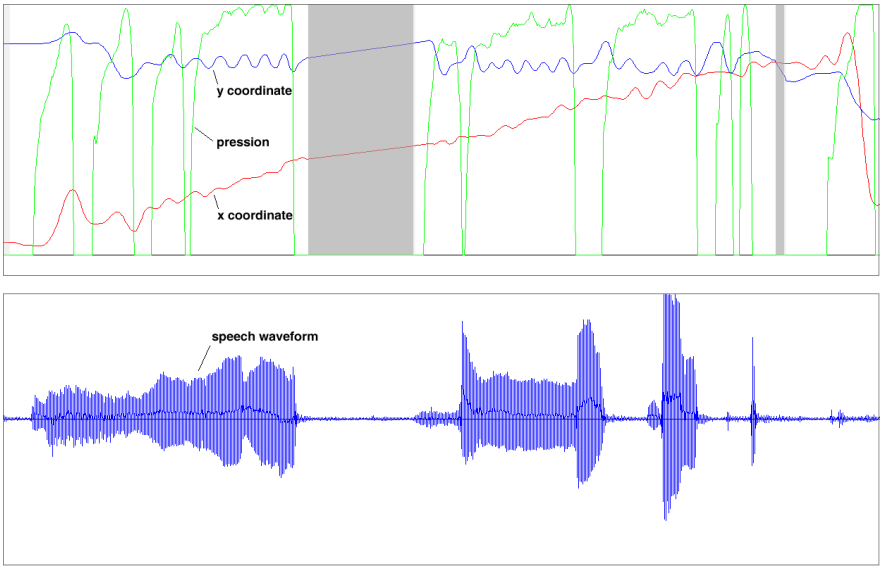


Fig. 1. Synchronized visualization of handwriting and speech signals. Azimuth and elevation angles are not displayed for sake of clarity. The upper part of the graph shows the evolution of x and y coordinates and the pression p. The bottom part shows the speech amplitude. On this visualisation, all signals are synchronized thanks to the timestamps.

2.2 Comments on CHASM Signature Data

We performed a visual inspection of CHASM signatures for several different users. Strokes and acoustic events are of course not always synchronized in the same way. In most of the cases, a given acoustic event either is synchronized with the stroke either starts slightly after the stroke (see Fig. 1). In some realizations,

the speech event starts slightly before the stroke. The average synchronization times correspond roughly to syllables when the signature contains clear sequences of letters, which is the case for most of the signatures. These observations are in accordance with the acquisition protocol of MyIDea where the subjects were asked to speak in such a way that the sounds correspond roughly in time with the written symbols.

Flourishes are usually present in signatures, most frequently at the end of signatures. When flourishes are happening, a large majority of users are not producing acoustic events on top of them (as illustrated on the last stroke in the example of Fig. 1). If the signature contains only flourishes or non-readable signs, the subject was simply asked to utter his name while signing. In this case, there is no specific synchronization of acoustic and stroke events.

In our previous work [6], we report on a usability survey conducted on the subjects that took part to MyIDea recordings. The main conclusions of the survey are the following. First, all recorded users were able to perform the signature acquisition. Speaking and signing at the same time did not prevent any acquisition to happen. Second, the survey shows that simultaneous acquisitions are acceptable from the user point of view.

2.3 Evaluation Protocols

In MyIDea, six *genuine* CHASM signatures are acquired for each subject per session. This leads to a total of 18 true acquisitions after the three sessions. After acquiring the genuine signatures, the subject is also asked to imitate six times the signature of another subject. Imitations are performed by letting the subject having an access to the *static* image and to the *verbal content* of the signature to be forged. In other words, access to the voice recording is not given to perform the forgery. This procedure leads to a total of 18 *skilled forgeries*[1] after the three sessions, i.e. six impostor signatures on three different subjects.

CHASM signature assessment protocols have been defined on MyIDea [6]. The protocols have been crafted to be as realistic as possible and to put in evidence difficulties tied to time variability. Two protocols have been defined:

- **Without time variability.** For each subject in the database, models are built using three spoken signatures sampled randomly out from the six genuine accesses of the first session. For testing, the three remaining signatures of the first session are used. The same procedure is repeated for sessions two and three, leading to a total of 70 users * 3 accesses * 3 sessions = 630 *genuine* tests. Two kinds of impostor attempts are considered: *random forgeries* and *skilled forgeries*. In the case of random forgeries, impostor attempts are performed using one signature for each of the remaining subjects

[1] The term *skilled forgeries* is used here to somehow comply with the nomenclature used in the literature about signature verification systems. However, one should note that there is no trained imitation of the speech signal as only the content is reproduced with no intentions to imitate the genuine voice. For the speech part, the term *passive* or *content-based* forgeries could be then more adequate.

in the database, giving a total of 70 users * 69 accesses * 3 sessions = 14490 random forgeries. In the second case, the 18 available skilled forgeries are used against each user, giving a total of 70 users * 18 accesses * 3 sessions = 3780 skilled forgeries.

– **With time variability.** For each subject, the six signatures from the first session are used to build the models. Genuine tests are performed on the six signatures of session two and session three, giving a total of 70 users * 12 accesses = 840 genuine tests. Random and skilled impostor attempts are performed in the similar manner as for the protocol *without time variability* with the distinction that models are here trained on the first session only, giving a total of 70 users * 69 accesses = 4830 random forgeries and 70 users * 18 accesses = 1260 skilled forgeries.

The amounts of tests mentioned above are approximative as some users did not complete all sessions.

3 System Description

We have chosen to use standard GMMs to model independently both streams of data, followed by a simple fusion at the score level (see Fig. 2). While this system uses straightforward feature extraction and modelling, it will allow us to validate the evaluation protocol and to draw our first conclusions regarding the impact of time-variability and skilled vs random forgeries. Performances are also measured on the speech stream alone (1), the signature stream alone (2) and on the fused systems (3).

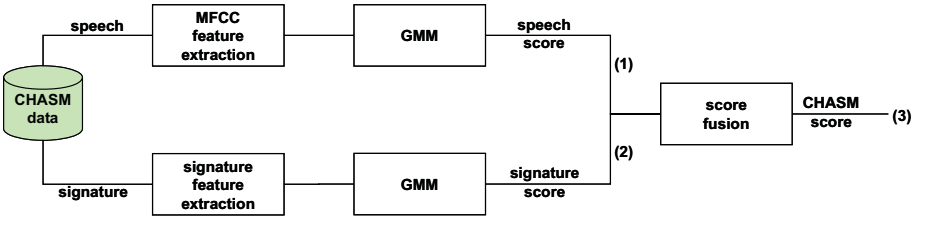


Fig. 2. Baseline CHASM signature verification system

3.1 Signature Features

For each point of the signature, we extract 25 dynamic features in a similar way as in [12]:

– the absolute speed and acceleration, the speed and acceleration in x and y directions and the tangential acceleration
– the angle α of the absolute speed vector, its cosine and sine, the derivative of α and its cosine and sine

- the pressure and the pressure derivative
- the azimuth and elevation angles of the pen and their derivatives
- the curvature radius
- the normalized coordinates $(x(n) - x_g, y(n) - y_g)$ relatively to the gravity center (x_g, y_g) of the signature
- the length to width ratio of windows of 5 and 7 points centered on the current point and the ratio of the minimum over the maximum speed on a window of 5 points centered on the current point.

The features are mean and standard deviation normalized on a per user basis.

3.2 Speech Features

We use Mel Frequency Cepstral Coefficients (MFCC) as features [14]. The frontend's frame size is 25.625 ms and the frame shift is 10 ms. The frontend extracts 12 MFCC coefficients and the energy. An energy-based speech detection module based on a bi-Gaussian model is applied to remove the silence from the data. MFCC coefficients are mean and standard deviation normalized using normalization values computed on the speech part of the data. We performed experiments including delta and delta-delta coefficients without further improvements of thr results. These features were then left apart in our baseline configuration for which results are reported here.

3.3 GMMs System

GMMs are used to model the likelihoods of the features extracted from the signature and from the speech signal. One could argue that GMMs are actually not the most appropriate models in this case as they are intrinsically not capturing the time-dependant specificities of signatures. HMMs would be potentially more adequate in this case. However, GMMs have been reported to compare reasonably well to HMMs in terms of signature verification [17] and are often considered as baseline systems in speaker verification. Furthermore, GMMs are well-known flexible modelling tools able to approximate any probability density function. With GMMs, the probability density function $p(x_n|M_{client})$ or *likelihood* of a D-dimensional feature vector x_n given the model of the client M_{client}, is estimated as a weighted sum of multivariate Gaussian densities (see e.g. [15]):

$$p(x_n|M_{client}) = \sum_{i=1}^{I} w_i \mathcal{N}(x_n, \mu_i, \Sigma_i) \tag{1}$$

in which I is the number of mixtures, w_i is the weight for mixture i and the Gaussian densities $\mathcal{N}$ are parameterized by a mean $D \times 1$ vector μ_i, and a $D \times D$ covariance matrix, Σ_i:

$$\mathcal{N}(x_n, \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\left(-\frac{1}{2}(x_n - \mu_i)'\Sigma_i^{-1}(x_n - \mu_i)\right) \tag{2}$$

In our case, we make the hypothesis that the features are uncorrelated and we use diagonal covariance matrices. By making the hypothesis of observation independence, the global *likelihood* score for the sequence of feature vectors, $X = \{x_1, x_2, ..., x_N\}$ is computed with:

$$S_c = p(X|M_{client}) = \prod_{n=1}^{N} p(x_n|M_{client}) \tag{3}$$

The likelihood score S_w of the hypothesis that X is **not** from the given client is here estimated using a world model M_{world} or *universal background model* trained by pooling the data of many other users. The likelihood S_w is computed in a similar way, by using a weighted sum of Gaussian mixtures. The optimal decision whether to reject or to accept the claimed user is performed comparing the ratio of client and world score against a global threshold value T. The ratio is often computed in the log-domain with:

$$R_c = \log(S_c) - \log(S_w) \tag{4}$$

The training of the client and world models is performed with the Expectation-Maximization (EM) algorithm [1] that iteratively refines the component weights, means and variances to monotonically increase the likelihood of the training feature vectors. The client and world model are trained independently by applying iteratively the EM procedure until convergence is reached, typically after few iterations. In our setting, we apply a simple binary splitting procedure to increase the number of Gaussian components to a predefined value. The world model is trained by pooling all the available genuine accesses in the database. The skilled forgeries attempts are excluded for training the world model as it would lead to optimistic results. Ideally, a fully independent set of users would be preferable, but this is not possible considering the small number of users (≈ 70) available.

3.4 Score Fusion

In our baseline system, we fuse the two modalities by simply summing the signature and the speech log-likelihood ratios with $R_{c,CHASM} = R_{c,speech} + R_{c,signature}$ which is a reasonable procedure if we assume that the local observations of both sub-systems are independent. This is however clearly not the case as the users are intentionally trying to synchronize their speech with the signature signal. Time-dependent score fusion procedures or feature fusion followed by joint modelling would be more appropriate than the approach taken here. These approaches are part of our future work. Also, more advanced score recombination or classification strategies could also be applied such as, for example, using a weighted sum of the likelihood or using classifier-based score fusion [7]. However, such fusion methods require parameters estimation on an independent development set which is currently not available. We will then report here fusion results using the simple summation as described above. We also report results using a *z-norm* score normalization preceding the summation. The z-norm is

here applied globally on both speech and signature scores, in a user-independent way. Such a normalization procedure makes sense if $R_{c,speech}$ and $R_{c,signature}$ are distributed according to a Gaussian distribution. As the mean and standard deviation of the z-norm are estimated a posteriori on the same data set, z-norm results are unrealistic but give however an optimistic estimation of what could be the performances.

4 Results

Results of biometric systems are classically measured in terms of impostor False Acceptation FA and client False Rejection FR error rates that vary as a function of the decision threshold T. Operating points (FA,FR) can then be plot on a (x,y) figure with T as parameter. Detection Error Tradeoff (DET)[13] plots are often used in which the x and y axis follow a normal deviate scale. If the scores are normally distributed the DET curve will be close to a straight line, enabling easy observation of system contrasts. We also report our results in terms of Equal Error Rates (EER) which are obtained for $FA = FR$.

For all the results reported here, we have used 64 mixtures in our world models. Experiments with different world model sizes have been conducted, leading to the conclusion that 64 mixtures give good results for all protocols. This number could probably be different if more or less training data would be available. Table 1 shows the evolution of the EER as a function of the number of mixtures in the client models, using protocol *with time variability and random forgeries*. We tested with 8, 16, 32, 64 and 128 Gaussian mixtures and concluded that, on average for all users, the optimal number of mixtures lie around 16 mixtures. Similar conclusions were obtained for the other protocols. For an improved version of the system, one could compute an optimized number of Gaussian mixtures for each user and modality such as taking into account the length of the signatures [11] or computing a cost functions that balances modelling errors and model complexity [17]. In the rest of this section, we report results with 16 Gaussian mixtures for the client models.

Table 1. Equal Error Rate (EER) as a function of the number of Gaussian mixtures in the client models, protocol with time variability

number of mixtures	8	16	32	64	128
signature	7.3	6.4	7.4	8.6	11.0
speech	12.0	12.2	14.1	14.8	15.0
sum fusion	4.7	4.7	5.8	6.8	8.0

Figure 3 illustrates the DET curve of the speech system, the signature system and the z-norm fusion of both systems for the protocol *with* (left part) and *without* (right part) time variability, and using random forgeries. We can conclude from this figure that the speech modelisation performs better than the signature

for single session experiments. However, when multi-session accesses are considered, signature performs better than speech. Signature and speech modalities suffer from time-variability but in different degrees. The speech modality seems to be more sensitive to time variability than the signature modality. The z-norm fusion brings a clear amelioration of the results for both protocols.

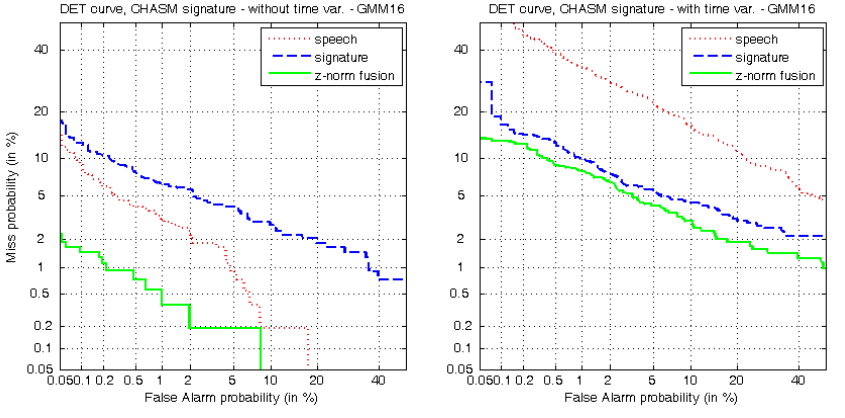


Fig. 3. DET curve - fusion of the signature and speech GMM systems, protocol without time variability (left) and with time variability (right), random forgeries

Table 2 summarizes the results in terms of ERR for the different protocols. The following conclusions can be drawn. For the skilled forgeries protocol, a drop of 3.3% of absolute performance is observed (9.4 - 6.1) when testing on signatures acquired in different sessions as the enrollment. For the speech modality, the impact is even more important with an absolute decrease of about 15% of the EER (19.5 - 3.7). Such a drop in the performance can be due to several factors. First, the modelling technique we used may not be robust enough against time variabilities. Using a MAP adaptation of the world model to build client models would be a potential amelioration in this regards. Second, it is probable that users show a larger intra-variability for the speech than for the signature modality. Third, the speech modelisation may suffer from variabilities of the acquisition conditions: different position of the headset-mounted microphone, environmental noise, etc., while the signature acquisition is more stable.

Another conclusion is that skilled forgeries decreases systematically and significantly the performance in comparison to random forgeries. For the protocol *with time variability*, a drop of almost 100% relative performance is observed for the signature modality and about 50% for the speech modality. We have to note again here that the forger do not try to imitate the voice of the user but actually say the genuine verbal content.

The sum fusion, although very straightforward, brings systematically a clear improvement of the results. These results are in favor of the CHASM

methodology. Interestingly, the z-norm fusion is better than the sum fusion for the protocol without time variability and is worse in the case of the protocol with time variability. A visual analysis of the score distribution of both modalities, before z-norm and after z-norm, lead us to a potential intuitive interpretation of this behavior. The application of the z-norm is, by nature, aligning the score distributions of both modalities. While this is good to fuse scores that lies in different ranges, the z-norm is also giving equal importance to each modalities. This is of course not favorable in the case of systems showing very different individual performances.

Table 2. Protocol with and without time variability, 16 Gaussian mixtures for the client GMMs, 64 mixtures for the world

time variability	without (%EER)		with (%EER)	
forgeries	random	skilled	random	skilled
signature	4.0	6.1	5.3	9.4
speech	2.0	3.7	14.0	19.5
sum fusion	**1.7**	**3.1**	**3.5**	**6.9**
z-norm fusion	**0.6**	**1.3**	**4.1**	**8.7**

5 Conclusions and Future Work

A baseline verification system using GMMs for modelling CHASM signatures has been presented. Results obtained with this system show that the use of both modalities outperforms these modalities used alone. Results also show that there is a clear impact of time variability and skilled forgeries on the performances. In our future work, we plan to investigate the use of more robust modelling techniques against time variability and forgeries. In this direction, we have identified potential modelling techniques such as MAP adaptation of the world GMMs, user-dependent model order, HMMs, time-dependent score fusion, fusion at the feature level followed by joint modelling, etc. As soon as an extended set of CHASM signature data will be available, experiments will be conducted according to a development/evaluation set framework. Another part of our future work will be to investigate CHASM handwriting to build verification systems.

Acknowledgments

This work was mainly supported by the Swiss National Science Foundation program "Interactive Multimodal Information Management (IM2)", as part of NCCR and by the EU BioSecure NoE project. We warmly thank Asmaa El Hannani for her precious feedbacks when we were experimenting with GMMs based systems.

References

1. A.P. Dempster, N.M. Laird, and Rubin D.B. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society*, 39(1):1–38, 1977.
2. B. Dumas et al. Myidea - multimodal biometrics database, description of acquisition protocols. In *In proc. of Third COST 275 Workshop (COST 275)*, pages 59–62, October 27 - 28 2005. Hatfield (UK).
3. M. Fuentes et al. Identity verification by fusion of biometric data: On-line signature and speech. In *Proc. COST 275 Workshop on The Advent of Biometrics on the Internet*, pages 83–86, November 2002. Rome, Italy.
4. S. Garcia-Salicetti et al. Biomet: a multimodal person authentication database including face, voice, fingerprint, hand and signature modalities. In *4th AVBPA*. Springer-Verlag, 2003.
5. J. Hennebert, B. Dumas, C .Pugin, F. Evéquoz, A. Humm, D. Petrovska-Delacrétaz. MyIDea home page. http://diuf.unifr.ch/go/myidea, 2005.
6. A. Humm, J. Hennebert, and R. Ingold. Combined handwriting and speech modalities for user authentication. Technical Report 06-05, University of Fribourg, Department of Informatics, 2006.
7. A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38:2270–2285, 2005.
8. S. Krawczyk and A. K. Jain. Securing electronic medical records using biometric authentication. In *Audio- and Video-based Biometric Person Authentication (AVBPA)*, pages 1110–1119, Rye Brook, NY, 2005.
9. F. Leclerc and R. Plamondon. Automatic signature verification: the state of the art–1989-1993. *Int'l J. Pattern Recog. and Artif. Intel.*, 8(3):643–660, 1994.
10. L. Lee, T. Berger, and E. Aviczer. Reliable on-line human signature verification systems. *IEEE Trans. Pattern Anal. and Mach. Intel.*, 18(6):643–647, June 1996.
11. B. Ly-Van, R. Blouet, S. Renouard, S. Garcia-Salicetti, B. Dorizzi, and G. Chollet. Signature with text-dependent and text-independent speech for robust identity verification. In *Proc. MMUA*, pages 13–18, December 2003.
12. B. Ly Van, S. Garcia-Salicetti, and B. Dorizzi. Fusion of hmm's likelihood and viterbi path for on-line signature verification. In *Biometrics Authentication Workshop*, May 15th 2004. Prague.
13. A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The det curve in assesment of detection task performance. In *Eurospeech 1997*, pages 1895–1898, Rhodes, Greece, 1997.
14. L. Rabiner and B.-H. Juang. *Fundamentals Of Speech Recognition*. Prentice Hall, 1993.
15. D. Reynolds. Automatic speaker recognition using gaussian mixture speaker models. *The Lincoln Laboratory Journal*, 8(2):173–191, 1995.
16. D. Reynolds. An overview of automatic speaker recognition technology. In *Proc. IEEE ICASSP*, volume 4, pages 4072–4075, 2002.
17. J. Richiardi and A. Drygajlo. Gaussian mixture models for on-line signature verification. In *Proc. 2003 ACM SIGMM workshop on Biometrics methods and applications*, pages 115–122, 2003.
18. C. Vielhauer. *Biometric User Authentication for IT Security*. Springer, 2006.

Kalman Tracking with Target Feedback on Adaptive Background Learning

Aristodemos Pnevmatikakis and Lazaros Polymenakos

Athens Information Technology, Autonomic and Grid Computing,
Markopoulou Ave., 19002 Peania, Greece
{apne,lcp}@ait.edu.gr
http://www.ait.edu.gr/research/RG1/overview.asp

Abstract. This paper proposes novel algorithms and system architecture for tracking targets in video streams. The proposed system comprises a variation of Stauffer's adaptive background algorithm with spacio-temporal adaptation of the learning parameters and a Kalman tracker in a feedback configuration. In the feed-forward path, the adaptive background module provides target evidence to the Kalman tracker. In the feedback path, the Kalman tracker adapts the learning parameters of the adaptive background module. The proposed feedback architecture overcomes the problem of stationary targets fading into the background, commonly found in variations of Stauffer's adaptive background algorithm and is capable of automatic initialization without the need for an initial background image.

1 Introduction

Target tracking in video streams has many applications, like surveillance, security, smart spaces [1], pervasive computing, and human-machine interfaces [2] to name a few. In these applications the targets are either human bodies, or vehicles. The common property of these targets is that sooner or later they exhibit some movement which is evidence that distinguishes them from the background and identifies them as foreground targets.

The segmentation of foreground objects can be accomplished by processing the difference of the current frame from a background image. This background image can be static [3] or can be computed adaptively [4]. The drawback of the static background image is that background does change. In outdoor scenes natural light changes and the wind causes movement of trees and other objects. In indoor scenes, artificial light flickers and pieces of furniture are moved around. All such effects can be learned by an adaptive background algorithm [5] and any of its modifications, like [6,7]. Such an algorithm detects targets as segments different from the learned background, but depends on the targets' movement to keep a fix on them. If they stop, the background learning process fades them into the background.

Once a target is initialized, a tracking system should be able to keep a fix on it even when it remains immobile for some time. In this paper, we propose a novel tracking system that addresses many of the above mentioned limitations by utilizing a feedback mechanism from the tracking module to the adaptive background module

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 114 – 122, 2006.

which in turn provides the evidence for each target to the tracking module. We control the adaptive background parameters on a pixel level for every frame (spacio-temporal adaptation), based on a prediction of the position of the target. Under the assumption of Gaussian-like targets, this prediction can be provided by a Kalman filter [8].

This paper is organized as follows: In section 2 the adaptive background, measurement and Kalman tracking modules are detailed. Some results are presented in section 3. Finally, in section 4 the conclusions are drawn, followed by some indications for further work.

2 Tracking System

The block diagram of the tracking system is shown in Figure 1. It comprises three modules: adaptive background, measurement and Kalman filtering. The adaptive background module produces the foreground pixels of each video frame and passes this evidence to the measurement module. The measurement module associates the foreground pixels to targets, initializes new ones if necessary and manipulates existing targets by merging or splitting them based on an analysis of the foreground evidence. The existing or new target information is passed to the Kalman filtering module to update the state of the tracker, i.e. the position and size of the targets. The output of the tracker is the state information which is also fed back to the adaptive background module to guide the spacio-temporal adaptation of the algorithm. In the rest of the section, we present the three modules in detail.

2.1 Adaptive Background Module

The targets of the proposed system (vehicles and humans) are mostly moving. The changes in the video frames due to the movement are used to identify and segment the foreground (pixels of the moving targets) from the background (pixels without movement). If a background image were available, this segmentation is simply the difference of the current frame from the background image. The foreground pixels thus obtained are readily grouped into target regions. A static image of the empty scene viewed by the camera can be used for background [3]. Unfortunately this is not practical and adaptive background approaches are adopted [4-7] primarily for two reasons: First, such an empty scene image might not be available due to system setup. Secondly and most importantly, background (outdoors and indoors) also changes: Natural light conditions change slowly as time goes by; the wind causes swaying movements of flexible background object (e.g. foliage); fluorescent light flickers at the power supply frequency; objects on tabletops and small pieces of furniture are rearranged and projection areas display different content. All these changes need to be learnt into an adaptive background model.

Stauffer's adaptive background algorithm [5] is capable of learning such changes with so different speeds of change by learning into the background any pixel, whose color in the current frame resembles the colors that this pixel often has. So no changes, periodic changes or changes that occurred in the distant past lead to pixels that are considered background. To do so, a number of weighted Gaussians model the appearance of different colors in each pixel. The weights indicate the amount of time

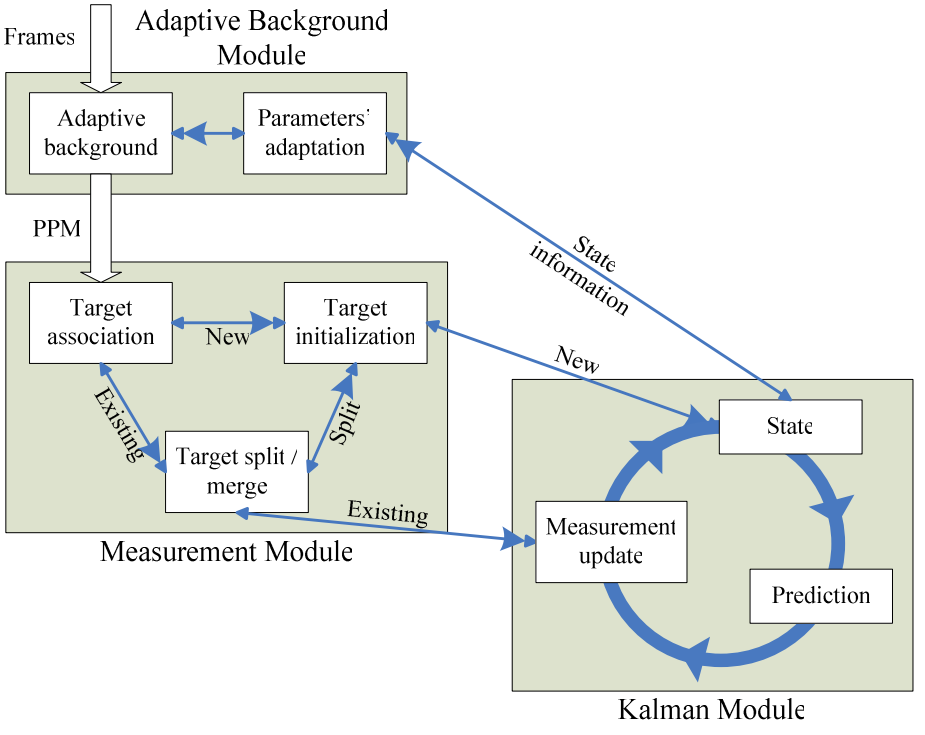


Fig. 1. Block diagram of the complete feedback tracker architecture

the modeled color is active in that particular pixel. The mean is a three dimensional vector indicating the color modeled for that pixel, while the covariance matrix indicates the extend around the mean that a color of that pixel is to be considered as similar to the one modeled. Colors in any given pixel similar to that modeled by any of the Gaussians of that pixel lead to an update of that Gaussian, an increase of its weight and a decrease of all the weights of the other Gaussians of that pixel. Colors not matching any of the Gaussians of that pixel lead to the introduction of a new Gaussian with minimum weight. Hence the possible updates of the weight of the i-th Gaussian of the pixel located at (x, y) at time t are

$$w_i(x,y,t) = \begin{cases} a & \text{new Gaussian} \\ (1-a)\, w_i(x,y,t-1) & \text{non-matching Gaussians} \\ (1-a)\, w_i(x,y,t-1) + a & \text{matching Gaussians} \end{cases} \quad (1)$$

where a is the learning rate.

Some variations of the Stauffer algorithm found in the literature deal with the way covariance is represented (single value, diagonal of full matrix) and the way the mean and covariance of the Gaussians are updated [6]. Some further variations of the algorithm address the way the foreground information is represented. The original algorithm and most of the modifications lead to a binary decision for each pixel: foreground or background [5,6]. In [7], the Pixel Persistence Map (PPM) is used

instead. This is a map of the same dimension as the frames with a value at each location (x, y) equal to the weight of the Gaussian matching the current color of the pixel at (x, y). Small PPM values indicate foreground objects, while large indicate background. The foreground/background threshold is left unspecified though.

The drawback of all the existing variations of Stauffer's algorithm is that stationary foreground objects tend to fade in the background with rate a. Small rates fade foreground objects slowly, but are also slow in adapting to the background changes, like the motion of a chair. Large rates favor background adaptation but tend to fade a target into the background when it stops. This fading progressively destroys the region of the tracked object, deforms its perceived shape and finally leads to loosing track of the object altogether. When the target resumes moving, foreground pixels will be marked only at the locations not previously occupied by the stationary target. When the target has fairly uniform coloration, this can lead to track loss even in the presence of movement.

We propose a feedback tracking architecture in order to addresses these problems. The threshold PPM serves as target evidence to the Kalman tracker. The state of the Kalman tracker contains the ellipse that describes every target. The learning rate is modified in elliptical regions around these targets. Thus instead of a constant value, a spacio-temporal adaptation of the learning rate is used:

$$a(x,y,t) = \begin{cases} \text{large if } (x,y) \text{ not near target at time } t \\ \text{small if } (x,y) \text{ near target at time } t \end{cases} \quad (2)$$

This delays fading of the targets and depending on the selection of the small learning rate and the motion of the targets can be sufficient. In some cases though where targets stay put for very long periods, even the small learning rate will gradually fade them into the background. If this starts happening (the target becomes smaller while its mobility is small), the normal weight update mechanism of (1) is bypassed. The weight of the current Gaussian is decreased and that of all the rest is increased with a rate that is inversely proportional to the mobility of the target, as this is estimated from the state of the Kalman tracker for this particular target. This fading prevention mechanism is not always in effect; it is only activated when targets are small and rather immobile, since the tampering of the weights is very forceful and affects the whole elliptical disk around the target, regardless if the pixel is actually foreground or not.

The second major proposed modification of Stauffer's algorithm addresses extreme flickering situations often encountered in night vision cameras. In such scenes the PPM needs to be bounded by a very low threshold in order not to consider flickering pixels as foreground. The threshold on the other hand tends to discard actual foreground pixels as well. The proposed solution is to adapt the threshold T in a spacio-temporal fashion similar to the learning rate in (2). i.e.

$$T(x,y,t) = \begin{cases} \text{small if } (x,y) \text{ not near target at time } t \\ \text{large if } (x,y) \text{ near target at time } t \end{cases} \quad (3)$$

This way flickering pixels are avoided far from the targets, while the targets themselves are not affected. The penalty of this strategy is the delayed detection of new very small targets.

These proposed feedback mechanisms on the learning rate lead to robust foreground regions regardless of the flickering in the images or the lack of target mobility, while they do not affect the adaptation of the background around the targets. When such flickering and mobility conditions occur, the resulting PPM is more suitable for target region forming that the original version of [7]. The forming of target regions is the goal of the measurement module, detailed next.

2.2 Measurement Module

The measurement module finds foreground segments, assigns them to known targets or initializes new ones and checks targets for possible merging or splitting. The information for new targets or targets to be updated is passed to the Kalman module.

The measurement process begins by processing the adaptively thresholded PPM to obtain foreground segments. This involves shadow detection based on [9], dilation, filling of any holes in the segments and erosion. The obtained segments are checked for possible merging based on their Mahalanobis distance and are further considered only if they are large enough. These segments are associated to targets based on their Mahalanobis distance from the targets. Non-associated segments generate new target requests to the Kalman module.

The targets are subsequently checked for possible merging based on how similar they are. Since we are using a Kalman tracker, the targets are described by two-dimensional Gaussians [8]. If two such Gaussians are too similar, the targets are merged. Finally, very large targets are checked for splitting. This is necessary as, for example, two monitored people can be walking together and then separate their tracks. Splitting is performed using the *k*-means algorithm on the pixels of the foreground segment comprising the target. Two parts are requested from the *k*-means algorithm. These parts are subsequently checked to determine if they are distinct. For this, the minimum Mahalanobis distance of the one with respect to the other is used. If the two parts are found distinct, then they form two targets. The one part of the foreground evidence is used to update the existing target, while the other part is used to request a new target from the Kalman tracker.

2.3 Kalman Tracking Module

The Kalman module maintains the states of the targets. It creates new targets should it receive a request from the measurement module and performs measurement update based on the foreground segments associated to the targets. The states of the targets are fed back to the adaptive background module to adapt the learning rate and the threshold for the PPM binarization.

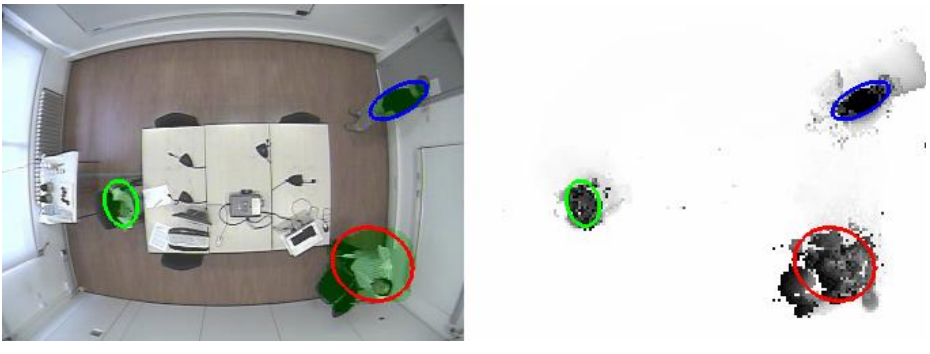
Every target is approximated by an elliptical disc, i.e. can be described by a single Gaussian. This facilitates the use of a Kalman tracker. The target states are seven-dimensional; they comprise of the mean of the Gaussian describing the target (horizontal and vertical components), the velocity of the mean (horizontal and vertical components) and the three independent terms of the covariance matrix.

The prediction step uses a loose dynamic model of constant velocity [10] for the update of the mean position and velocity. As for the update of the three covariance terms, their exact model is non-linear, hence cannot be used with the Kalman tracker; instead of using linearization and an extended Kalman tracker, the covariance terms

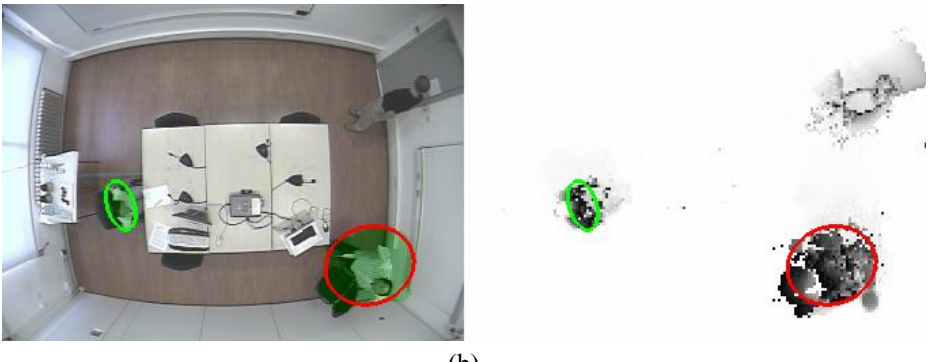
are modeled as constant. The variations of the velocity and the covariance terms are permitted by the state update variance term. This loose dynamic model permits arbitrary movement of the targets. It is very different to the more elaborate models used for tracking aircraft. Aircraft can perform a limited set of maneuvers that can be learned and be expected by the tracking system. Further, flying aircraft can be modeled as rigid bodies thus strict and multiple dynamic models are appropriate and have been used extensively in Interacting Multiple Model Kalman trackers [11,12]. Unlike aircraft, street vehicles and especially humans have more degrees of freedom for their movement which includes apart from speed and direction changes obstacles arbitrarily, rendering the learning of a strict dynamic model impractical. A strict dynamic model in this case can mislead a tracker to a particular track even in the presence of contradicting evidence [13].

3 Experimental Results

The proposed feedback tracking architecture is tested on the CLEAR evaluations (video sequences coming from the CHIL and VACE projects). In this section we



(a)



(b)

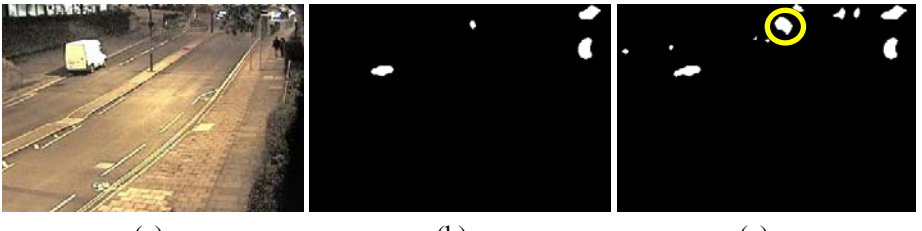
Fig. 2. Tracking with (a) and without (b) the proposed feedback that spacio-temporally adapts the learning rate of the adaptive background module. Without the proposed scheme, the top-right target is lost entirely, whereas the center-left one is in the process of fading (see the PPM on the right column of the figure). The moving bottom-right target is not affected.

show the effect of the algorithm on the data, more specifically, how it is applied successfully both in indoor and outdoor environments. Figures 2 and 3 show the effect of the spacio-temporal learning rate adaptation to the PPM when a target remains stationary. When the proposed adaptation is not used, stationary targets fade, so that the system either looses track (Figure 2) or has reduced tracking accuracy (Figure 3).

(a) (b)

Fig. 3. Tracking with (a) and without (b) the proposed feedback that spacio-temporally adapts the learning rate of the adaptive background module. Without the proposed scheme, the stationary target is no longer tracked. Instead, the system tracks the chair the target has started moving.

Figure 4 shows the effect of the spacio-temporal adaptation of the threshold for the binarization of the PPM in the adaptive background module. The morphological processing of the thresholded PPM can result in false alarms if the threshold is not adapted by the states of the Kalman tracker.



(a) (b) (c)

Fig. 4. Tracking in outdor night video (a) with (b) and without (c) the proposed feedback that spacio-temporally adapts the threshold for the binarization of the PPM in the adaptive background module. Without the proposed scheme, night camera flicker generates false alarm segments, one of which exceeds the size threshold and initiates a false target (marked by the yellow circle).

4 Conclusions

The proposed tracking architecture of the adaptive background and the Kalman tracking modules in a feedback configuration combines the immunity of Stauffer's algorithm to background changes (like lighting, camera flicker or furniture movement), with the stability of the targets of the static background, no matter if they

move or not. Utilizing the Kalman tracker, gates are effectively built around the tracked targets that allow association of the foreground evidence to the targets.

The use of Kalman filtering necessitates the approximation of the tracked objects by two-dimensional Gaussians. This can be troublesome depending on the nature of the objects and the camera. Gaussians are sufficient approximations to vehicles. They are also sufficient approximations to human bodies when the camera viewing conditions are far-field (fisheye ceiling or road surveillance cameras) and the dynamic model of the Kalman tracker is loose. The limbs lead to important deviations from the Gaussian model for close viewing conditions. In such conditions, multiple occluding targets are common and the loose dynamic model is no longer capable of tracking. To overcome the problem of many, occluding, non-Gaussian-like targets, future extensions of the proposed tracking architecture will replace the Kalman tracker with CONDENSATION [14] algorithm. In a second extension of the proposed tracker, occlusions can also be handled if multiple synchronized and calibrated cameras [15] are available, to allow three-dimensional models of the targets.

Acknowledgements

This work is sponsored by the European Union under the integrated project CHIL, contract number 506909. The authors wish to thank the organizers of the CLEAR evaluations and acknowledge the use of data coming from the VACE project for testing the algorithm.

References

[1] Waibe1, H. Steusloff, R. Stiefelhagen, et. al: CHIL: Computers in the Human Interaction Loop, *5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Lisbon, Portugal, (Apr. 2004).

[2] Pnevmatikakis, F. Talantzis, J. Soldatos and L. Polymenakos: Robust Multimodal Audio-Visual Processing for Advanced Context Awareness in Smart Spaces, *Artificial Intelligence Applications and Innovations*, Peania, Greece, (June 2006).

[3] H. Ekenel and A. Pnevmatikakis: Video-Based Face Recognition Evaluation in the CHIL Project – Run 1, *Int. Conf. Pattern Recognition*, Southampton, UK, (Mar. 2006), 85-90.

[4] McIvor: Background Subtraction Techniques, *Image and Vision Computing New Zealand*, (2000).

[5] Stauffer and W. E. L. Grimson: Learning patterns of activity using real-time tracking, *IEEE Trans. on Pattern Anal.and Machine Intel.*, 22, 8 (2000), 747–757.

[6] P. KaewTraKulPong and R. Bowden: An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection, in *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems (AVBS01)*, (Sept 2001).

[7] J. L. Landabaso and M. Pardas: Foreground regions extraction and characterization towards real-time object tracking, in *Proceedings of Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI '05)*, (July 2005).

[8] R. E. Kalman: A New Approach to Linear Filtering and Prediction Problems, *Transactions of the ASME – Journal of Basic Engineering*, 82 (Series D), (1960) 35-45.

[9] L.-Q. Xu, J. L. Landabaso and M. Pardas: Shadow Removal with Blob-Based Morphological Reconstruction for Error Correction, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (March 2005).

[10] S.-M. Herman: A particle filtering approach to joint passive radar tracking and target classification, PhD thesis, University of Illinois at Urbana-Champaign, (2002), 51-54.

[11] H. A. P. Bloom and Y. Bar-Shalom: The interactive multiple model algorithm for systems with Markovian switching coefficients, *IEEE Trans. Automatic Control*, 33 (Aug. 1988), 780-783.

[12] G. A. Watson and W. D. Blair: IMM algorithm for tracking targets that maneuver through coordinated turns, in *Proc. of SPIE Signal and Data Processing of Small Targets*, 1698 (1992), 236-247.

[13] Forsyth and J. Ponce: Computer Vision - A Modern Approach, Prentice Hall, (2002), 489-541.

[14] M. Isard and A. Blake: CONDENSATION - conditional density propagation for visual tracking, *Int. J. Computer Vision*, 29, (1998), 5-28.

[15] Z. Zhang: A Flexible New Technique for Camera Calibration, Technical Report MSR-TR-98-71, Microsoft Research, (Aug. 2002).

Da Vinci's Mona Lisa
A Modern Look at a Timeless Classic

Dennis Lin[1], Jilin Tu[1], Shyamsundar Rajaram[1],
Zhenqiu Zhang[1], and Thomas Huang[1]

Beckman Institute, University of Illinois Urbana Champaign, Urbana, IL

Abstract. There has been some controversy over the true subject portrayed in Leonardo da Vinci's Mona Lisa. In particular, there are suggestions that the painting is in fact a self-portrait. In this paper, we analyze the shapes of the features in the Mona Lisa and a known self-portrait of Leonardo da Vinci using active shape models. We conclude that the two faces have very distinct features and that they appear to be of different genders.

1 Introduction

The Mona Lisa by Leonardo da Vinci is arguably one of the most famous pieces of art in the world. In [1], Schwartz argues that the surface painting, supposedly of Lisa Gherardini, is actually a self-portrait of Leonardo. She supports her claim by noting that the eyes, nose-tip, and mouth of the Mona Lisa lines up with a known self-portrait of da Vinci. However, she does not take into account the possible range of face shape variation in the human population. We attempt a more sophisticated analysis by constructing an active shape model [2] (ASM) of human faces. These models capture a mean shape and can describe the correlation in the ways that the shape vary. In addition to using a more advanced model, we expect a more reliable result because we are using more landmarks. Our results contradict that of Schwartz and suggest that the two images feature distinct faces.

2 Analysis

One of the first things that we noticed when we began our analysis is that the Mona Lisa has cracks which may obscure its texture; the da Vinci self-portrait is lacking in texture altogether. Thus, we use only shape information in our comparison. To accomplish this, we built an ASM face model, which will tell us about the natural range of variation in face shapes. We also tuned a k-nearest neighbor gender classifier to check the shapes of Mona Lisa's features.

2.1 Database

We used a manually labeled database of 488 frontal faces of different ethnicities. The database contains 151 females and 337 males. Each face was labeled with

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 123–128, 2006.

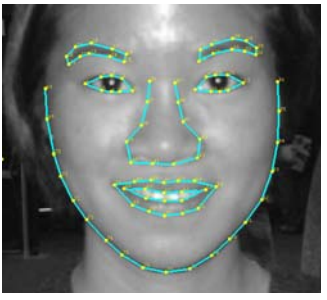


Fig. 1. Sample image showing our landmarks

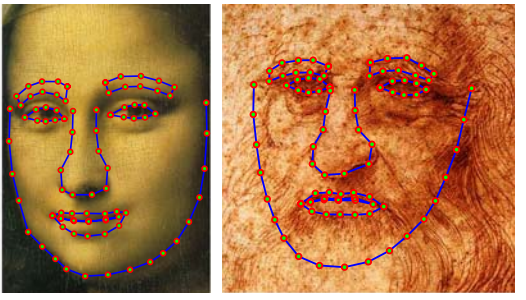


Fig. 2. Our labeling of the Mona Lisa and Leonardo da Vinci's self portrait

87 landmarks, as shown in Figure 1. We also manually labeled the Mona Lisa image and the da Vinci self portrait, as shown in Figure 2.

2.2 Active Shape Model

Active shape models [2] (ASM) are an effective way to represent deformable objects such as faces. In this section we review the original ASM model as well as our modifications of that algorithm.

Preprocessing. Although similar in spirit, our preprocessing steps differ significantly from those specified in [2]. In his paper, Cootes distinguished among three types of landmarks. Type 1 landmarks include eye and lip corners, which have semantic meaning and are relatively easy to localize. For faces, type 2 landmarks would include the points along the bottom of the nose because they correspond to regions of high curvature. Finally, we have many type 3 points around the face boundary which are interpolated from type 1 and type 2 points. The inconsistencies of manual labeling causes these points to slide along the curve, dramatically increasing variance. We took several steps to reduce this effect. First, we aligned on the eye corners instead of doing a least-square solution on all of the points as suggested by the original paper. We also fit a cubic spline to the edge and resampled the points so that they are more equally spaced, as shown in Figure 3. Finally, we enforced symmetry by averaging the left and right halves of the face.

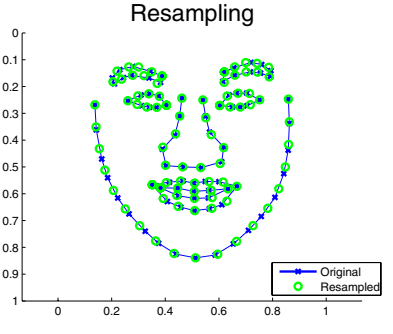


Fig. 3. An example of resampling the the landmarks

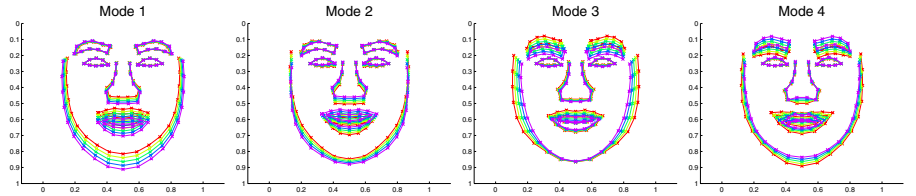


Fig. 4. Plots of $\bar{\mathbf{x}} + \alpha_j \mathbf{v}_j$ for $-3\sqrt{\lambda_j} \leq \alpha_j \leq 3\sqrt{\lambda_j}$ and $j \in \{1, 2, 3, 4\}$. These plot represent three standard deviations in either direction from the mean along the first four modes of variation.

Principal Component Analysis. At the heart of the active shape model is the application of principal component analysis (PCA). This process fits a low-dimensional ellipsoid to the data, providing an estimate of the amount and direction of variations.

The analysis begins by finding the mean shape $\bar{\mathbf{x}}$, given by

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i, \tag{1}$$

where $\mathbf{x}_i$ is one of N faces in our database. The next step is to build a scatter matrix $\mathbf{S}$, which is given by

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T. \tag{2}$$

Now, let λ_j, $\mathbf{v}_j$ be the corresponding the j^{th} largest eigenvalue and corresponding unit eigenvector of $\mathbf{S}$. That is, $\mathbf{v}_j$ and λ_j satisfy

$$\mathbf{S}\mathbf{v}_j = \lambda_j \mathbf{v}_j, \ \mathbf{v}_j^T \mathbf{v}_j = 1. \tag{3}$$

Each of the eigenvectors represents a mode of variation, or a way that a particular face shape can differ from the mean face shape. The first four modes are shown in Figure 4.

Usually, PCA is used as a dimension-reducing technique that simplifies and smoothes the data. To do this, we throw away the modes with low eigenvalues. In order to guide how many modes to keep, we note that retaining only the k most significant modes will allow us to capture a fraction of the total variance given by

$$\tau = \frac{\sum_{j=1}^{k} \mathbf{v}_j}{\sum_j \mathbf{v}_j}. \tag{4}$$

Choosing $\tau = 0.95$, we find that we need to use $k = 12$ eigenvectors. This is in contrast to the 35 that would be necessary had we not resampled, or the 21 that would be necessary if we did not enforce symmetry.

Mahalanobis Distance. The result of the PCA analysis is an efficient representation of face shapes. We can now describe a face $\mathbf{x}_i$ with twelve coefficients $x_{i,1}, \ldots, x_{i,12}$ given by

$$x_{i,j} = \mathbf{x}_i \mathbf{v}_j. \tag{5}$$

This set of coefficients indicates how our particular face differs from the mean face along the major modes of variation. They minimizes the error in the reconstruction

$$\mathbf{x}_i \approx \bar{\mathbf{x}} + \sum_{j=1}^{12} x_{i,j} \mathbf{v}_j. \tag{6}$$

In addition, we can compare two shapes by evaluating the Mahalanobis distance between the sets of coefficients. Suppose shape $\mathbf{x}$ is described by $x_1, \ldots, x_{12}$ and shape $\mathbf{y}$ is described by $y_1, \ldots, y_{12}$. Then, the Mahalanobis distance between the two is given by

$$\|\mathbf{x} - \mathbf{y}\|^2_{\text{Mahalanobis}} = \sum_{j=1}^{12} \frac{(x_j - y_j)^2}{\lambda_j}. \tag{7}$$

This distance takes into account how much variation there is along a particular mode. That is, this metric reflects the fact that two shapes which differ along a mode that normally has a low variance are more dissimilar than two shapes that differ along a mode that usually has a high variance. Therefore, it is a good measure for determining the relative distance between two face shapes given the range of variations that occur in our database.

2.3 Gender Classifier

Our database contained gender labels, so in addition to simply performing shape analysis, we we built a k-nearest-neighbor classifier to perform gender classification. If Schwartz's claims are true, we may expect to classify both Mona Lisa and Leonardo da Vinci as male. A k-nearest-neighbor classifier does not require explicit model building, but we must select an appropriate k. To do this, we tried different values and measured the error rate using the leave-one-out test methodology. The results are shown in Figure 5. The error rate consistent for varying k, showing that it is in fact possible to separate gender using face shapes. The minimum was at $k = 19$, which achieves 79% accuracy on our database. This is the value we used for our experiment.

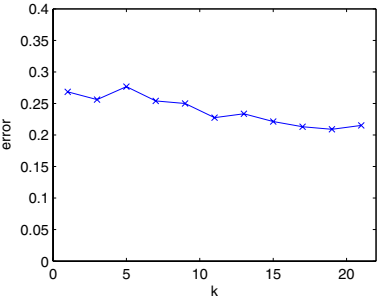


Fig. 5. Error rate of k-nearest-neighbor classification as function of k

3 Results

The Mahalanobis distance between Mona Lisa and Leonardo da Vinci is 4.02. Figure 6 plots this against the distances between points in our database. We note that the Mona Lisa-Leonardo da Vinci distance is over 3.6 standard deviations

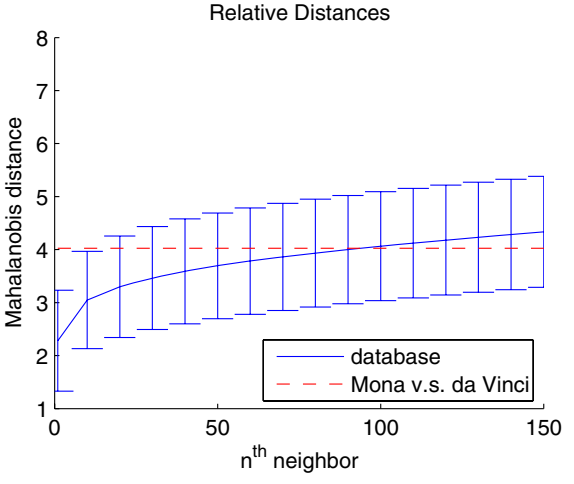


Fig. 6. Distribution of distances in our database. The plot shows the Mahalanobis distance from a point to its n^{th} nearest neighbor. The line is the mean distance, and the error bars represent a 95% confidence interval. The dashed line is the distance between Mona Lisa and Leonardo da Vinci.

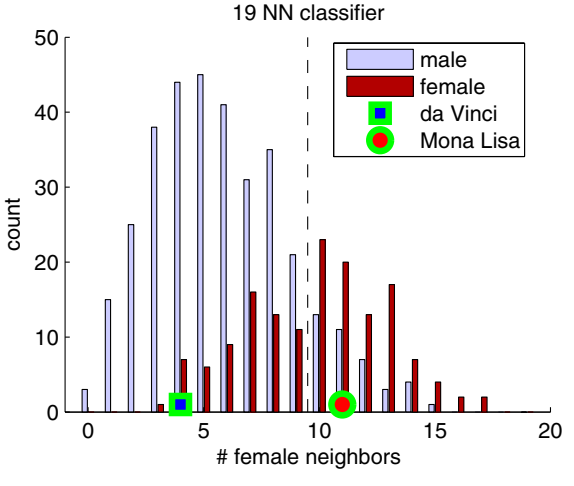


Fig. 7. Summary of 15-nearest neighbor classification results. Points to the right of the dotted line are classified as female. The histogram summarizes the result of on our labeled database. The square and circle indicates the ranking of the Leonardo da Vinci face and the Mona Lisa face, respectively.

away from the mean of adjacent faces in our database. This indicates that it is unlikely that they come from the same face. Indeed, the distance between the two is comparable to a point and its 95^{th} closest neighbor, indicating that they are quite distinct.

Applying our gender classifier to the Leonardo da Vinci and Mona Lisa faces yields the result detailed in Figure 7. Those to the right of the dotted line were classified as female. The histograms show the distribution of male and female faces during the leave-one-out testing. As expected, the bulk of the female faces are to the right of the line, while the majority of the male faces are to the left. Not surprisingly, Leonardo da Vinci has 15 male neighbors out of 19, placing it solidly in the male camp. Mona Lisa, however, had 11 female neighbors and was classified as female. This is consistent with our results above and with our conclusion that the to faces are distinct.

4 Discussion and Future Work

We have shown that the appearance of Mona Lisa is very strongly distinct from that of Leonardo da Vinci. This result contradicts that of Schwartz's previous work. However, we believe that our results are more reliable because we use a larger number of feature points. We also characterized the differences between the two depictions relative to the normal variance found in human faces. The relatively enormous distance between the two face shapes suggests that they are different. The fact that our classifier find that the Mona Lisa is female also bolsters our claim.

In the end, there is a limit in applying techniques designed for photographs on pieces of art. The shapes rendered by an artist's hand is subject to interpretation and style. Without asking him personally, it would be very difficult to determine if Leonardo da Vinci intended to paint a (perhaps feminized) version of himself when he create the Mona Lisa. However, we hope that the publication of this result will encourage researchers in computer vision to examine this issue. In particular, we recognize that our analysis is entirely two dimensional. An extension into 3D may better model the pose variations seen in the paintings.

Acknowledgments

The development of the algorithms used in this study was supported in part by ARDA/DOT and in part by Yamaha Motor Co. The application of these algorithms to the Mona Lisa/Da Vinci problem is supported by internal funding of the Beckman Institute.

References

1. Schwartz, L.F.: Morphing the three faces of mona: the decision-making steps leonardo used to create his mona lisa. Computers and Graphics **19**(4) (1998) 529–539
2. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models–their training and application. Comput. Vis. Image Underst. **61**(1) (1995) 38–59

The Connector Service-Predicting Availability in Mobile Contexts

Maria Danninger[1], Erica Robles[2], Leila Takayama[2], QianYing Wang[2], Tobias Kluge[1], Rainer Stiefelhagen[1], and Clifford Nass[2]

[1] Universität Karlsruhe, InterAct Research,
Am Fasanengarten 5, 76131 Karlsruhe, Germany
{maria,kluge,stiefel}@ira.uka.de
[2] Stanford University, CHIMe Lab,
94305 Stanford, USA
{ewoka,takayama,wangqy,nass}@stanford.edu

Abstract. In this thriving world of mobile communications, the difficulty of communication is no longer contacting someone (the receiver), but rather contacting them in a socially appropriate manner. Ideally, senders should have some understanding of a receiver's availability in order to make contact at the right time, in the right contexts, and with the optimal communication medium. This paper describes our ongoing research on the Connector, an adaptive and context-aware service designed to facilitate efficient and appropriate communication. We describe a set of empirical studies whose results converge upon the important subject of people's availability in mobile contexts.s

1 Introduction

1.1 Project CHIL and the Connector Service

Computers are becoming more ubiquitous and seamlessly integrated into everyday life. At present, considerable human attention is devoted to operating and attending to computers, and people are often forced to spend precious time fighting with technologies rather than engaging in human interaction and communication. This unfortunate trend moves us further away from Mark Weiser's motivation for the post-PC era of ubiquitous computing, getting us away from staring at PC monitors with computers at the center of attention in order to re-engage in human interaction [1].

Having computers anticipate our needs and provide us with relevant information and services would help people to break the technological attention barrier and re-engage in meaningful human interactions. Such human-centered computational tools would be particularly beneficial in meeting situations or technologically-mediated communication.

Within the framework of the **CHIL project - Computers in the Human Interaction Loop -** we intend to develop context-aware, proactive computer services that assist people during daily interactions with others [2]. Rather than expecting people spend their time attending to technology, CHIL's goal is to develop computer services that are sensitive in attending to human activities, interactions, and intentions. In order to act

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 129–141, 2006.

in a proactive yet implicit way, services should be able to identify and possibly even understand human activities.

In this paper, we describe our ongoing research on a CHIL service called **the Connector** [3]. The Connector is designed to intelligently connect people at the right place, the right time, and with the best possible medium for socially appropriate communication.

1.2 Availability in Mobile Contexts

Modern communication technologies bring considerable advantages, as well as burdens, to both the sender and the receiver in a communication [4]. Despite the fundamentally social nature of communication, research and design of communication technologies disproportionately favors the initiators of communication, the sender, over the target of communication, the receiver. Therefore, the guesswork involved in making decisions about how and when to contact someone is placed in the hands of the sender. The sender calls when their situation is conducive to communication, but they do so with little knowledge of the receiver's situation. The problem is further exacerbated with the advent of mobile communication which decouples location from situation, thus decreasing the capacity for a sender to make informed decisions about the person they are calling. In the past, people were called at locations which reasonably described their current activity e.g. home, work, or school, but now that mobile phones are anywhere that people are, little contextual information can be inferred about the state of the receiver.

If there is no need to communicate in a synchronous way, this problem is much less apparent. Asynchronous communication, such as email, is reasonably convenient since the sender worries less about disturbing the receiving party. Instant messaging clients let the receiver set one's own online availability status, which has a number of benefits. However, the growing use and constant attending to instant messages often becomes a distraction to users [5] [6]. Moreover, text-based communication lacks the emotional richness and nuance found in oral communication where the same phrase said differently means differently. IM users are obviously aware of this pitfall as they very often use it to negotiate availability for a phone conversation [7].

The Connector aims at empowering both the receiver and the sender to establish communication, either synchronous or asynchronous, in a contextually appropriate way based on each party's availability. In order to inform the development of this technology, we have conducted a series of studies designed to understand how current mobile phone users negotiate and decide upon when to engage in communication. The results from these studies inform the development of a model of availability for communication and this model facilitates the design of the Connector communication service.

The remainder of this paper is organized as follows. Section 2 is an overview of related work. Section 3 outlines a series of large-scale field studies with one hundred mobile phone users conducted on a university campus to understand patterns of mobile phone use in everyday life. Section 4 describes the design and prototype implementation of the Connector Service as an adaptive and multimodal communication tool, with front-end clients running on smart phones, standard phones, WinXP and the World Wide Web. The following section presents results from a pilot study on availability collected with this system. We end with a summary and conclusions.

2 Previous Work

It is assumed that 50 per cent of phone communications fail because they do not happen in the right moment in time [8]. Brown and Randell [9], in their essay on context sensitive telephony, discussed the possibility of an automated agent that blocks calls on the behalf of users. They concluded that a better solution would be to provide the callee's context information to the caller to let the caller make a more informed decision about whether or not to initiate the call.

A number of mobile awareness systems are doing work that aligns well with this approach. Context Phone [8] is a Smart phone application which enables users to share their context with their others who use the same application. Both "Awarenex" [10] and "Live Addressbook" [11] are systems on mobile devices that allow users see others' location and availability status with an interface similar to today's instant messaging buddy lists. Users can consider this information in order to make more informed decisions about contacting others. The "Live Contacts" system [12] also provides preferences for communication channels. "Enhanced Telephony" [13] is a desktop-based design of an enhanced PC-phone. In all of these systems, users must either manually update their availability state or context information is inferred automatically from sources such as login time, personal calendars, messenger status, idle time of computer input devices, and engagement in communication activities.

SenSay [14] is a mobile phone that follows a different approach. It adapts to changing user states by manipulating ringer volume, vibration, and phone alerts for incoming calls. SenSay uses a number of wearable sensors including accelerometers, light and microphones mounted on the user's body to provide context information.

The Connector is designed to combine many of the features mentioned above. Additionally, the Connector leverages machine-learning techniques to sense the receiver's availability from automatically gathered context cues. Connector clients run on Smart phones as well as WinXP platforms; it supports a standard phone dialogue interface. To inform the design of the Connector, we conducted a suite of large-scale field studies in order to understand mobile usage patterns in terms of receiver availability. These field studies consisted of both an exploratory survey field study a field experiment that involved controlled, randomly assigned experimental conditions.

3 Large-Scale Mobile Phone Field Studies

We ran large-scale field studies with approximately one hundred mobile phone users in order to understand and enhance our understanding of mobile phone usage patterns. We designed the studies with a special emphasis on receivers in everyday life.

The **first study** focused on revealing the contextual characteristics that correspond to successful mobile connections. Multiple methods of inquiry were employed in order to provide a better understanding of receiver availability across a diversity of contexts within which mobile communication occurs. This model is carried forward into the **second study**, where it informs the design of a basic Connector service that facilitates contextually appropriate mobile phone conversations.

The overall intent of the field studies was to discover how to best facilitate successful, efficient, socially appropriate communication through mobile phone technology

for the Connector service. Analysis of the extremely large amounts of collected data points is our current work in progress. We present our initial results in the following sections.

3.1 Availability Study – Everyday Mobile Phone Usage Patterns

This study investigated the contextual circumstances under which successful, missed, and rejected calls occurred. The study deployed a system capable of randomly pinging users throughout the day to determine availability for conversation in situ. Additionally, participants indicated their availability for mobile conversations using an online calendar, hosted by an Exchange server. Availability probes were deployed throughout a period lasting one full week. The length of this period means allows data to be analyzed for differential use patterns during weekdays versus weekends, daytime versus evenings etc. Each ping consisted of a call, deployed by the server, to a human receiver. If the receiver answered, the server played a recorded voice prompt, asking the receiver to indicate his or her current availability for a conversation (by voice or DTMF). Upon being pinged, receivers responded with their availability by hitting a key on their mobile phone keypad, 1-9, regarding their availability at the moment. Our telephony server logged whether or not the appointments scheduled in the receiver's calendar, whether or not the receiver answered, and ambient acoustic noise during the call.

Finally, at the close of each day, participants completed an online questionnaire about their context when each phone call was received. They describe features of the situation that influenced their decision to communicate, or not, at the time. For example, *"I was visiting with a friend. We were talking and not too busy but he is a very close friend,"* or, *"My boss was in the room and asked what the call was about*," or insights like *"I realized that when filling out the online calendar, I did not always block off times I could not answer the phone but instead times when I did not want to answer the phone."*

This study allows multiple evaluation strategies: correlation between plans and situated availability (similar to Suchman's plans and situated action [15]), models of usage patterns based on time of day or calendared activities, and content analysis of contextual features that predict availability for communication. The study will provide an empirical foundation for deriving a model of receiver availability. This model will be carried forward into the design of the Connector, in order to facilitate contextually appropriate mobile phone conversations.

3.2 Connector Study – Mobile Phone Communication with Connection Assistance

The second study was designed to examine how the Connector, by facilitating a conversation between two individuals, affects ease of communication, social judgments and perceptions of each other, and assessments of the Connector system, across both coordination and collaboration tasks. Students engaged in both individual- and group-centered activities involving two features of the Connector. Feature 1 was the 1:1 Connector service, which facilitated 1:1 connections between two individuals by offering callee availability information to callers at the time of the call. Feature 2 was the 1:N Connector service, which facilitated connections from one caller to the a group of in-

dividuals, offering connections to those group members who are currently available at the time of the call.

Students used their own mobile phones to call the Connector telephone server (see Section 4). The system encouraged senders to complete a call only when receivers were available, thereby minimizing the risk of inappropriate interruptions or missed calls. Additionally, by placing a system between the sender and receiver, both sides were free to provide detailed information about their availability without allowing direct surveillance by other humans. Eight teams of students were asked to complete coordination and a collaboration tasks with eleven or twelve teammates. Teams were arranged to maximize likelihood of unfamiliarity with teammates by choosing students from different discussion sections. Experimental conditions were randomly assigned.

In a between-subjects 2 (feature 1 or no feature 1) x 2 (feature 2 or no feature 2) design, we had four conditions in this study with two teams per condition. We varied two dimensions, directly informed by the two Connector features, 1:1 Connector feature and 1:N Connector feature. The control condition consisted of team members completing the tasks through typical mobile communication. The other three experimental conditions consisted of either only the 1:1 Connector feature, only the 1:N Connector feature, or both the 1:1 and 1:N Connector features.

There were two experimental tasks in this study; one featured coordination and one featured collaboration. In the **first task**, participants had to contact at least half of their teammates for help solving a Mystery Person task. This task required collaboration and information exchange between participants because team members each had a different set of clues that collectively complete the process of elimination to identify their team's Mystery Person, but did not require a face-to-face meeting. In the **second task**, participants were asked to arrange a face-to-face meeting. At this face-to-face meeting, teammates took a group picture to prove that they met at the requested location (see Figure 1).

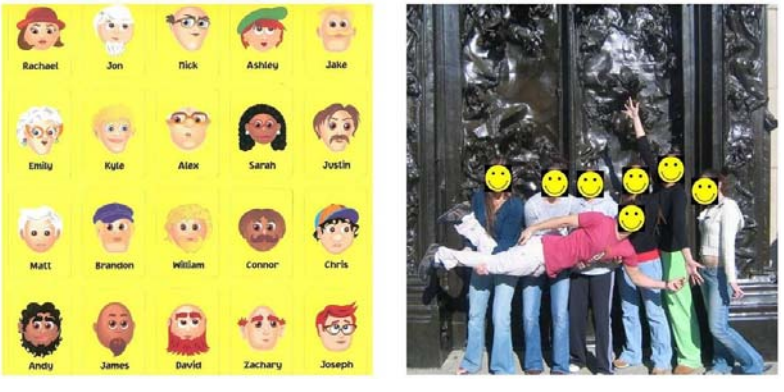


Fig. 1. Left: All possible faces in the Big Connector Study's Guess-Who-Task, each team had to collect clues from team members to find their Mystery Person. Right: Group picture taken during the coordination task.

3.3 Discussion

These large-scale studies were not only field studies, but also involve experimental manipulations that allow for controlled examination of what differences in Connector features will make a difference for future users of the system.

All studies were performed at Stanford University during the fall 2005 quarter, using roughly 100 college-aged students from mixed disciplines (social science, engineering, and humanities) from an introductory course in Communication. Evaluation of the collected data is work in progress. Initial results from the Availability Study indicate that participants with calendar appointments marked as busy or free did not significantly predict how available participants were for communication according to the in-the-moment availability as measured by pings from the server. This supports the claim that there is a problem with calendaring information, which could be framed as planned (scheduled in calendar) availability as being very different from situated (in-the-moment) availability. Therefore, we conclude that calendar information alone is not sufficient to estimate availability for communication.

This work suggests a need for a better predictive framework for receiver availability and a more detailed understanding of receiver location, environment, activities, social relationship to caller and communication urgency. The Connector service described in the following section is an attempt to create and test such a predictive framework.

4 Design and Implementation of the Connector Service

The Connector is an adaptive and context-aware service designed for efficient and socially appropriate communication. It maintains an awareness of its users' activities, preoccupations, and social relationships to mediate a proper moment and medium of connection between particular people. In this system, personal agents act as virtual administrative assistants, who know how to selectively facilitate some calls while blocking others.

In order to be ubiquitously accessible for users, Connector clients can run on a set of Smart phones, as a Windows XP application, and as a web service. A dialogue interface is supported for all standard phones. Machine-learning techniques are used in order to learn individual user availability, from automatically detected context cues, as well as direct user input.

4.1 System Overview

The system architecture in Figure 2 shows how various clients are integrated and communicate to the core Connector module. So far, all the logic is placed on the server side of the system. The core Connector module is responsible for collecting context information and learning the users' availability model. Data such as user preferences and settings are stored in a database. Calendar information is hosted by an Exchange server. The client-server communication is XML-based over TCP. Brief description of the clients follows.

4.2 Connector Clients

Connector **smart phones** run a custom-built graphical Connector user interface, indicating current receiver availability. The system controls incoming calls, outgoing calls,

messages, and phone alerts. Currently supported platforms include Sony Ericsson P900 Symbian phone and Windows CE devices. The smart phones contact the user's server placed personal agent to determine how to respond to incoming calls or messages. The communication takes place via Wifi or GPRS.

MyConnector is the Connector client running on **Windows XP** (screenshot shown in Figure 3). It provides an interface to set preferences and manage contacts. Along with phone communication, MyConnector lets the user send emails, send instant messages, and allows conference calls (via the Skype API). In the contact list, various symbols are displayed showing the availability of the contact person for communication media such as Skype IM, Skype call, email, office phone, home phone and cell phone. MyConnector does as well gather PC activity in the background as context cue to learn user availability.

Not everyone owns a smart phone and will be willing to install the MyConnector WinXP application. Therefore, a set of Connector functionality is available from every **standard phone** via a voice dialogue system. By calling a person's (toll-free) Connector number, the call is routed through our telephone server. As the caller, once you identify yourself and the person you want to contact, the Connector service will inform you about the receiver's current availability; and then proceed to route or block the call, accordingly. This is the setup that was used and tested in the field studies described in section 3.

User profiles and current availability are also viewable from **web browsers**. Each user has a public profile accessible by anyone and different custom profile, which typically has more detailed information for selected individuals. Figure 3 demonstrates a public profile. We integrated the Google Maps service to display the current location of a user. Also, an overall availability level and details of the callee's current location is

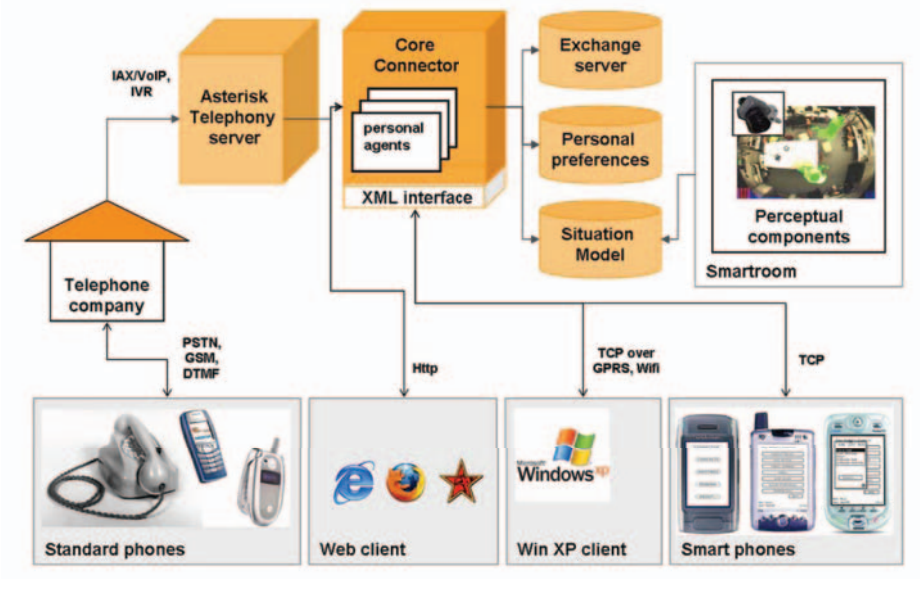


Fig. 2. Overall Connector architecture

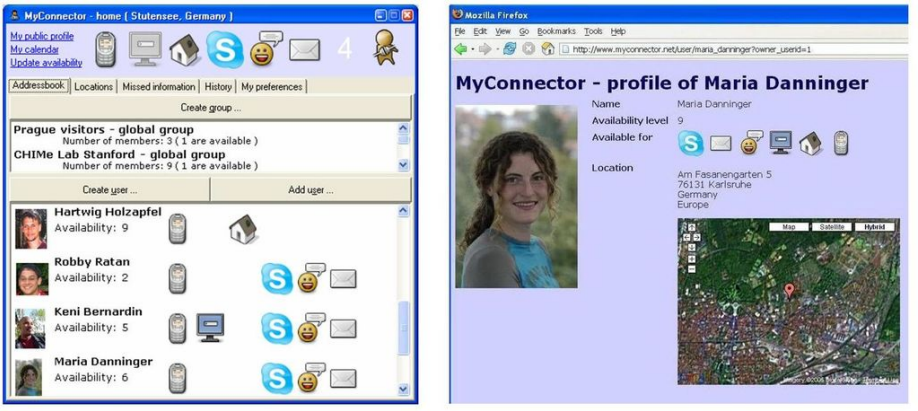


Fig. 3. Screenshot of the MyConnector Windows XP client and web interface showing a user's public profile

displayed. Icons indicate availability for different communication media; active icons may be clicked to use that communication medium to contact the person. The level of information granularity displayed is user-defined in the owner's privacy settings.

4.3 Privacy Settings

Whenever personal data such as this is broadcasted, privacy immediately becomes an extremely important issue to the user. This becomes obvious, as most people do not want their detailed location being shown in a Google map on the web. The Connector provides the opportunity to specify *who* should be able to see *what* information *when*. E.g. *I want all my colleagues to see the building I am in, but only during working hours, but my family can always see where I am. The default should be only specifying what world continent I am on (as opposed to what country, city, street,building, or room).*

We implemented hierarchical privacy rules in a rule based system. Each rule specifies when it will fire depending on the time of day (free time or work time) and the location of the user. Such privacy rules can be created for users or groups in the address book; the default setting is used for unknown persons.

According to previous research [16][17], it is necessary to provide appropriate default settings when it comes to privacy related data. We ran a survey with 43 people to find appropriate default privacy settings for the Connector service. In this survey people were asked which details about their location and current activity they would like to broadcast to their wife/husband, family, friends, acquaintances, coworkers and their boss, during work time and free time. The time of day seemed to be only relevant for work-related persons (co-workers, boss). As expected, less known persons (such as acquaintances) were less trusted than people in more proximate social circles (such as family and friends).

4.4 Learning User Availability

The Connector uses machine-learning techniques to model contextual knowledge about the user and to infer the user's availability for communication. Input comes from

automatically detected context cues collected in the MyConnector WinXP application such as:

- **personal calendars**: entries in a personal calendar
- **PC activity**: keyboard and mouse events, active application, window switching frequency
- **location in office**: based on the analysis of video-streams from cameras installed in our research labs

Our system uses Bayesian networks. Investigating different classifiers, such as decision trees and other Bayes-based classifiers, and attribute combinations is work in progress.

Further analysis of the MyConnector pilot study described in section 5 will definitely impact the design of future MyConnector systems as we learn more about the usefulness of various context cues to be used as attribute combinations for the classifier.

5 MyConnector Pilot Study-Predicting User Availability

In order to inform the development of the MyConnector technology, we have conducted an experiment to understand which or which combination of a large set of context cues (either automatically collected by the MyConnector system or manually entered by the participants) have a strong predictive power for gauging one's availability.

5.1 Study Design

We ran a pilot study with 9 participants at research labs in Karlsruhe and Stanford for one week in order to investigate the predictive power of context information currently used by MyConnector, as well as a number of possible future measures, that were self-reported by our subjects in this study. We used an experience sampling technique, and pinged subjects about their current availability and current activity during their normal daily activities. A popup window appeared on their screen about every 20 minutes. By simultaneously collecting sensor data as described in Chapter 4.4 we can examine offline which of the following factors would have produced the best estimates of one's availability.

Additionaly, participants were asked to manually enter availability feed-back every 20 minutes, **self-reported context cues** were the following:

- **current location**: e.g. office, home, transit on campus
- **accessibility of communication media** e.g. Email, IM, office phone
- **social acceptability**: How socially acceptable would it be to take a phone call in the current situation
- **activity category**: one of: *basic needs* (e.g. eating, sleeping), *household needs* (like cooking), *intellectual needs* (at job, at meeting, ...), *transportation needs*, *communication needs*, *interpersonal needs* (socializing with friends, ...), *personal needs* (reading, watching TV, ...)
- **mental and physical engagement**: while doing the current activity

- **importance and urgency**: of the current activity
- **point in lifespan**: of the current activity (beginning, middle or end)
- **collocation**: with how many people

Providing no data was interpreted as not available at all. **Ground truth** was a **self-reported availability level** between 1 and 4, meaning:

- **1**: not available at all (e.g. sleeping, swimming)
- **2**: basically not available, but exceptions possible (e.g. meeting, driving a car)
- **3**: busy but can be disturbed (e.g. internet browsing, preparing slides)
- **4**: free, communication encouraged (e.g. doing public transportation, waiting for an appointment)

For the offline data analysis, we used an iterative learning approach to get comparable results to an online classifier. Data entries were sorted by timestamps, and for each data entry t the classifier has been trained on data entries 1 to t-1. For the final result, the classification results for each item were counted.

5.2 Initial Results and Observations

Table 1 shows the results of using various context cues in the Bayes classifier to predict availability. We see that learning a person's availability seems to be a very hard task. Partially, this may be due to the fact that a person's 'stated' or 'planned' availability as e.g. scheduled in a calendar, does not always correspond to their 'demonstrated' or 'in-the-moment' availability. On the other hand, if an event is planned, interruptions are probably much more awkward then in a spontaneous meeting.

The time of day was especially powerful in combination with self-reported location information, but only for people with a structured day and regular office hours. As well, the predictive power of personal calendar information was only significant for some of

Table 1. Results of using various context cues in the Bayes classifier to predict availability

attribute	**classifier result** [%]	**attribute**	**classifier result** [%]
Time (hour, weekday)	54.3	Collocation with others	48.0
Location	51.6	Interaction with others	49.2
Time - Location	58.0	Activity category	42.8
Active program	48.4	Activity importance	47.3
Keyboard activity	28.5	Activity urgency	46.1
Mouse activity	29.3	Activity mental engagement	45.0
Window switching	28.7	Activity physical engagement	38.7
Online connection	46.9	Activity point in lifespan	37.7
Skype	32.3		
Active Program and Online Connection	49.1		

our participants. It was found that the existence of an appointment is not always a good indicator for a lower availability.

Results from PC activity information were lower then expected, even though the active program showed to be the best indicator. This was probably due to the fact that a number of participants used multiple computers throughout the day. People were more interruptible if engaged in Skype communication and in general, if they had connection to the internet.

Looking at a person's activity information, the urgency and importance as well as the mental engagement in the current activity seemed to be more valuable then the activities category and a person's physical engagement in the activity. This may be good news, since urgency and importance could be extracted from email communication or calendars entries, whereas engagement in the activity may be harder to sense. Our participants were more interruptible towards the end of an activity then at its beginning.

In order to estimate the statistical significace of some of the factors, we applied a standard multiple regression analysis to predict the dependent variables of level of availability set manually by the participants. Regression analysis shows that only two variables, the social acceptability of a phone call ($p \leq .01$) and the urgency of the current activity ($p \leq .01$), contribute significantly to predictions of the person's availability level. The R^2 for regression is significant ($R2 = .32, adjR2 = .31, F(1, 166) = 24.29$, $p \leq .01$). In other words, the less socially acceptable receiving a phone call in the current environment would be, the less available a users is. Also, participants were more available when doing current activities that were judged as not urgent. Along with these findings, physical engagement appears to be another possible predictor for availability (p=.078) as it approaches, but does not reach statistical significance. All other self-reported factors, such as the activity category, the point in lifespan of the activity, the importance of the current activity, and collocation with others are not statistically significant predictors of availability under this case ($p \geq .1$). We still believe that these factors may be interesting and aim to study a larger sample size to get a more thorough understanding of these factors and their predictive power for gauging one's availability.

This statistical analysis shows, that the results presented here should be considered with care. To strengthen some of our observations, a larger experiment would be necessary.

6 Summary and Conclusions

The focus throughout the paper is on how receivers in mobile contexts negotiate and decide upon when to engage in a communication, to generate a model of receiver availability and to design more efficient and socially appropriate communication services, such as the Connector service.

Large-scale user studies with about 100 mobile phone users suggest that planned availability as scheduled in a calendar is very different from situated, in-the-moment availability and that we need a better predictive framework for receiver availability. As such, the Connector service was introduced as an adaptive context-aware service designed for efficient and socially appropriate communication. Pilot studies with the Connector prototype attempted to show the impact of various contextual cues on the

user's availability. Results indicate that location and time, as well as the urgency of the current activity and social acceptability of a call in the current environment are significant indicators for a person's availability.

Further larger-scale studies are planned in order to get a more thorough understanding of these findings. All findings will be carried forward into the design of the future Connector service.

Ongoing work in Karlsruhe focuses on the development of perception technologies, such as audio-visual speaker tracking [18] and person identification, head pose estimation [19] and speech recognition [20]. Technologies will have to be improved and tuned to detect the most significant context cues automatically, in order to make the Connector a real proactive CHIL service.

Acknowledgements

This work has partially been funded by the European Commission under Project CHIL (http://chil.server.de, contract nr. 506909).

We thank Gopi Flaherty, Abhay Sukumaran, Shailo Rao, Anna Ho, Robby Ratan for their help and support in running the studies.

References

1. M. Weiser. The computer for the twenty-first century, Scientific American (1991), 94-100.
2. http://chil.server.de/
3. M. Danninger, G. Flaherty, K. Bernardin, H. K. Ekenel, T. Köhler, R. Malkin, R. Stiefelhagen, A. Waibel, The Connector - Facilitating Context-aware Communication, Proc. of the Int. Conf. on Multimodal Interfaces (ICMI), Trento, Italy, 2005.
4. C. Shannon and W. Weaver. The mathematical theory of communication. Urbana, IL: University of Illinois Press, 1962.
5. D. Avrahami and S. E. Hudson. QnA: Augmenting an Instant Messaging Client to Balance User Responsiveness and Performance. In Proc. of CSCW 2004, Chicago, USA, 2004.
6. E. Isaacs, A. Walendowski, S. Whittaker, D. J. Schiano and C. Kamm. The Character, Functions, and Styles of Instant Messaging in the Workplace. In Proc. of CSCW 2002, NY: ACM Press, 11-20. 2002.
7. B. Nardi, S. Whittaker and E. Bradner. Interaction and Outeraction: Instant Messaging in Action. In Proc. of the Conf. on Computer-Supported Cooperative Work (CSCW) 2000. NY: ACM Press, p. 79-88. 2000.
8. M. Raento, A. Oulasvirta, R. Petit, H. Toivonen. ContextPhone - A prototyping platform for context-aware mobile applications. In Proc. of IEEE Pervasive Computing, 4 (2): p. 51-59, Munich, Germany, 2005.
9. B. Brown, and R. Randell. Building a context sensitive telephone: some hopes and pitfalls for context sensitive computing. In CSCW Journal, special edition on context aware computing, 13, 3 (2004), 329-345.
10. J.C. Tang, N. Yankelovich, J. Begole, M. Van Kleek, F. Li and J. Bhalodia, ConNexus to Awarenex: Extending awareness to mobile users. In Proc. of the Conf. on Computer Human Interaction (CHI), New York, 2001.
11. A. E. Milewski, T. M. Smith, Providing presence cues to telephone users, Proc. of the Conf. on Computer-Supported Cooperative Work (CSCW), Philadelphia, Dec 2000.

12. Henri ter Hofte, Raymond A.A. Otte, Hans C.J. Kruse, Martin Snijders, Context-aware communication with Live Contacts, Proc. of the Int. Conf. on Computer-Supported Cooperative Work (CSCW), Chicago, Nov 2004.
13. J.J. Cadiz, A. Narin, G. Jancke, A. Gupta, M. Boyle, Exploring PC-telephone convergence with the enhanced telephony prototype. Proc. of the Int. Conf. on Computer Human Interaction (CHI), New York, USA, 2004.
14. D.P. Siewiorek et al., SenSay: A Context-Aware Mobile Phone, Proc. of the Int. Symposium on Wearable Computers (ISWC), White Plains, NY, Oct 2003.
15. L. Suchman. Plans and situated actions: The problem of human-machine communication. Cambridge: Cambridge University Press, 1987.
16. S. Patil, and A. Kobsa. Instant Messaging and Privacy. In Proceedings of HCI 2004, Leeds, U.K., pp. 85-88.
17. S. Lederer, J. Mankoff, and A. K. Dey. Who Wants to Know What When? Privacy Preference Determinants in Ubiquitous Computing. Short Talk in the Extended Abstracts of CHI 2003, ACM Conference on Human Factors in Computing Systems, pp. 724-725, April 5-10, 2003.
18. K. Nickel, T. Gehrig, R, Stiefelhagen, J. McDonough, A Joint Particle Filter for Audio-visual Speaker Tracking, In Proc. of the Int, Conference on Multimodal Interfaces (ICMI), Trento, Italy, October 2005.
19. M. Voit, K. Nickel, R. Stiefelhagen, Estimating the Lecturer's Head Pose in Seminar Scenarios - A Multi-view Approach, 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms - MLMI 2005, July 2005, Edingburgh, UK.
20. M. Wölfel, K. Nickel and J. McDonough, Microphone Array Driven Speech Recognition: Influence of Localization on the Word Error Rate, 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms - MLMI 2005, July 2005, Edingburgh, UK.

Multimodal Input for Meeting Browsing and Retrieval Interfaces: Preliminary Findings

Agnes Lisowska and Susan Armstrong

ISSCO/TIM/ETI, University of Geneva,
40 Blvd du pont d'Arve
1211 Geneva, Switzerland
{Agnes.Lisowska,Susan.Armstrong}@issco.unige.ch
http://www.issco.unige.ch

Abstract. In this paper we discuss the results of user-based experiments to determine whether multimodal input to an interface for browsing and retrieving multimedia meetings gives users added value in their interactions. We focus on interaction with the Archivus interface using mouse, keyboard, voice and touchscreen input. We find that voice input in particular appears to give added value, especially when used in combination with more familiar modalities such as the mouse and keyboard. We conclude with a discussion of some of the contributing factors to these findings and directions for future work.

1 Introduction

Several projects[1] involve, or have involved in the past, the collection of multimodal meeting data. In several of these, special SmartRooms have been designed in which meetings are recorded in such a way that the resulting multimodal meeting data can be easily synchronized, processed and stored. For example, in the IM2 project (in which the work presented here is grounded) meetings are recorded at the IDIAP SmartRoom [7] and the resulting data is stored in databases that contain video and audio tracks from a meeting, a text transcription of the meeting, as well as various levels of annotation, including linguistic (dialogue acts, topic segments, keywords) and meta-levels (meeting actions). Additionally, the meeting data contains electronic versions of all documents used in the meetings, copies of all notes taken by meeting participants, and what was written on the electronic whiteboard available in the room.

One of the central questions then, is how a real-world user such as an employee of a company where SmartRoom meeting data has been recorded can best exploit this data. Tucker and Whittaker [10] provide a good overview of the types of meeting browsers that have been developed in various projects and suggest a 4-category taxonomy for meeting browsers – audio, video, artifact and discourse. However, it appears that most if not all browsers that are described for the meeting domain rely on

[1] The *IM2* project http://www.im2.ch, the *AMI* project www.amiproject.org, *The Meeting Room Project* at Carnegie Mellon University, http://www.is.cs.cmu.edu/mie/, *Rich transcription of natural and impromptu meetings* at ICSI, Berkeley, http://www.icsi.berkeley.edu/icsiro.html, and the *Multimodal Meeting Manager* http://www.m4project.org/.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 142 – 153, 2006.

standard mouse and keyboard input. Little has been said about the possible benefits of incorporating multimodal input to a meeting browsing and retrieval system.

Early work on multimodal interaction involved interfaces incorporating speech, mouse and keyboard input [8]. More recently, work has focused on multimodal interaction involving spatio-temporal tasks such as plotting actions on a map, where pen and speech input are the dominant modalities [8]. In the latter, research showed that users do interact multimodally, and that in cases of voice-pen input, there seems to be a preference for specifying elements such as symbols, locations or physical objects using pen, while temporal and abstract concepts are more frequently expressed using voice [8]. Conversational speech input on its own has often been used in telephony based systems for contexts such as travel planning [3, 4], or in interactions with a virtual world such as the Hans Christian Anderson system [1], where children can 'talk' to a representation of the fairy tale author and ask about his life and work.

Due to the proliferation of window-based platforms, technologies such as the internet, and the commonality of input devices such as the mouse and keyboard, certain interaction paradigms seem to have asserted themselves in western computer culture. For example, the use of a mouse for direct manipulation of graphical objects on the screen (point-and-click browsing) or the use of the keyboard for targeted web-searching. Moreover, if and when language is used to seek out information, it is often via 'intelligent' keyword-driven searches, in which obtaining desired results quickly and efficiently requires a certain degree of skill and knowledge.

We maintain that in most office-type environments and for almost all office-type applications the mouse-keyboard paradigm will be strongly preferred and users will be reluctant to stray from it. However, we believe that meeting browsing and retrieval such as outlined above is a sufficiently new and different domain of interaction where users can be encouraged to try out and consistently use novel input modalities such as voice (including more complex natural language interaction), touchscreen or pen input. The difference in the multimedia meeting browsing and retrieval domain is not found at the level of the actual media artifacts that are stored in the database. The web contains examples of the same types of media (video, audio and text files) and users are perfectly content to use the mouse and keyboard to access them. Rather, the distinction is made at the underlying level – in the direct, though not necessarily explicit, relationships between the information contained *across* that media, and the elements of that information that a person would want to access.

Our assumption is that the results of the fuzzy underlying difference can best be exploited by providing the user with a variety of modalities with which to access the information, including the use of complex natural language through voice and/or keyboard input, in addition to pointing modalities such as the mouse and in the case of the experiments presented here, a touchscreen. It is important to note though, that in a system such as ours the spatio-temporal aspect similar to that investigated in previous multimodal work comprises only a small number of situations such as choosing the location or date of a meeting, while the presence of a graphical interface makes a conversational interface more complex than in telephony systems.

Together with colleagues at the Artificial Intelligence Laboratory at the Ecole Polytechnique Fédeéral de Lausanne, we have designed a system, Archivus [6], specifically to meet the needs of the multimedia meeting domain. The work in this paper describes the results from an experiment whose aim was to determine if multimodal

input provides an added value to interaction for the multimodal meeting browsing and retrieval domain, and if it does, what the nature of that interaction is. It is important to note here that we define 'added value' in terms of increased efficiency (number of pieces of information found in a given time), usefulness (whether the information *can* be found) and overall user satisfaction (subjective opinion from the user).

2 The Archivus System

The Archivus system, described in detail in [6], was designed based on indications from a user-requirements study for the meeting browsing and retrieval domain [5]. The requirements identified in this study gave indications of the types of things that people might want to know about meetings, which helped us structure both the layout of the interface elements and the paths which users would have to take to reach that information in the meeting database. Moreover, the system was designed to be flexibly multimodal, meaning that the user can use any of the input modalities available (in this case mouse, keyboard, touchscreen, or voice) alone or in any combination they choose. This allows us to study how people use multimodal input if there are minimal *a priori* constraints imposed on that use. The layout of the interface can be seen in Fig 1, where the different elements are numbered and described below.

To help users become familiar with the use of the system and the structure and content of the database, we chose to use a library or archive metaphor. In this metaphor, each meeting is represented by a book (see area 2 for an example) and the whole database of meetings by a bookcase (area 1). The user has a view of the bookcase at all times, and in particular in which meetings, and how many of them, the relevant information they are seeking can be found. When a user opens a book, they have access to all aspects of the meeting, including all media that is associated with it, be it audio, video, or textual. Users can browse these meetings, as they would a book, play audio/visual media (area 3), or search for particular elements in the meeting.

Users can specify search criteria in two ways. The first is through the user input bar (area 5) where they can ask for access to the information they are looking for using typed natural language. Language in this case can take the form of keywords or more complex linguistic expressions. However, even though users are allowed to ask free-form questions such as '*Who was late to the meeting on January 21st*?", the system does not perform question-answering. Instead, the system will show the user the areas of a meeting that could contain the answer that the user is looking for (in area 2). The second way of searching for specific information is by using the predefined criteria buttons (area 6). These buttons also serve as a representation of some of the underlying annotations in the database and users can point and click their way through these annotations down to the parts of an actual meeting

Users are also guided in their interactions by system advice (area 4) which gives them feedback about their search and hints about where the relevant information can be found. Finally, there is the current criteria list (area 7) which serves to remind the user of all of the criteria that they have given to the system up to that point in their interaction, and allows users to remove an undesirable criterion without having to re-specify others. This feature is particularly useful if the criteria have been generated as a result of natural language processing done by the system. General system buttons such as *Help* and *Reset* are presented in area 8.

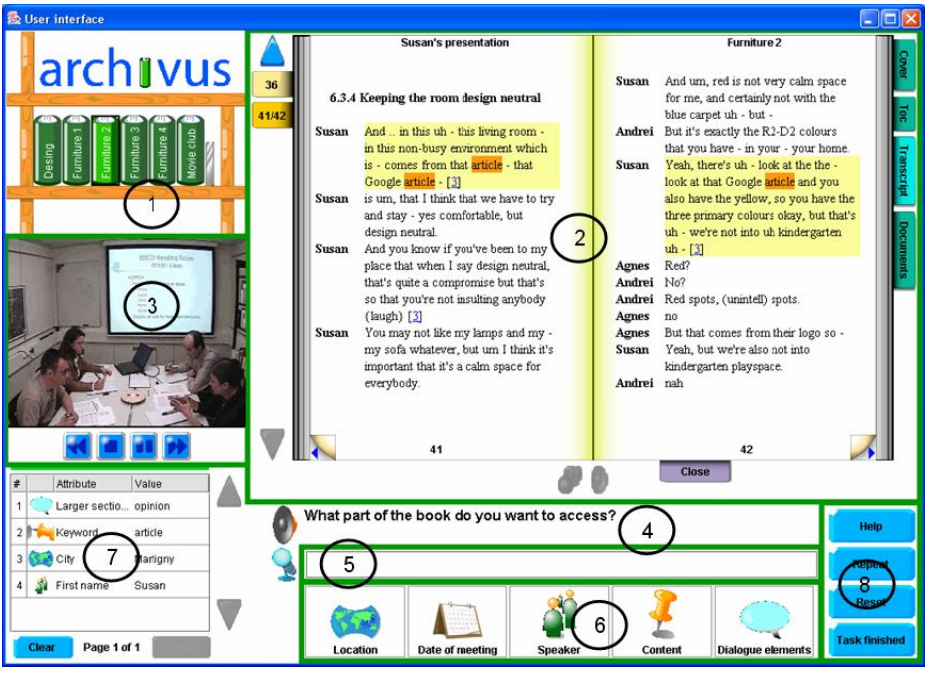


Fig. 1. The Archivus interface

3 The Experimental Environment

In our experiments, we are interested in the order and frequency of use of particular modalities, the language queries submitted by the user in their interactions, and the success rate under each experimental condition. In order to gather this information we record users as they interact with the system via two cameras (one that shows their face and another that provides a clear view on their hands and the input devices) and equipment that records everything that happens on the user's screen. The user is provided with a computer (in this case a desktop PC with a touchscreen), speakers, a wireless mouse and keyboard, and a lapel microphone.

As one of the goals of this work is to see whether natural language will bring added value to the interaction, we wanted to allow the use of natural language in the experiments without the overheard of implementing speech recognition and natural language processing at the early stages of testing (in part because it is hard to predict with a new domain what types of vocabulary and language processing capabilities would be necessary). Consequently, we decided to adopt the Wizard of Oz methodology [2, 9]. In Wizard of Oz experiments, a user interacts with what they believe to be a fully implemented system that uses natural language (and other modalities if needed). In reality, the natural language recognition and processing components are simulated by a human being, a 'wizard' in another room. The wizard hears what the user says and has a view of their interface, so that they can control the interactions.

In our experimental environment, the information from the face camera, the user's microphone and their screen is streamed directly to the wizard's room, where there are 2 computers (one for natural language processing and one that shows what is happening on the user's screen). An additional monitor shows the user's face as they are interacting with the system, which lets the wizard view their reaction to it. With this information, the wizard has a good overview of what the user is doing, which allows them to simulate the language processing as smoothly and accurately as could be reasonably expected with state-of-the-art systems.

Since in the future the wizard will be replaced by a speech recognizer and a natural language processing module, neither of which is likely to exhibit close to human performance, we purposely ask our wizards to be imperfect and make mistakes. These always involved 'recognition errors', where the wizard pretended to have misunderstood what the user was asking for and provided the 'wrong' response. Since almost all of the users as well as the wizard were non-native speakers of English and the cognitive load on the wizard was quite high to begin with, such errors happened naturally, and the wizard did not have to produce them on purpose very often. However, if the wizard produced a specific error during interaction with a particular user, care was taken to ensure that the error was reproduced if the situation in which it originally occurred happened again. Moreover, the wizard was asked not to over-interpret user input, in order for the simulated processing to be as close as possible to the realistic language processing capabilities foreseen for the system. This was facilitated by the fact that the wizard was the same for all participants and was the person responsible for the development of the natural language processing modules, and therefore has detailed knowledge of the existing and potential capabilities of the system.

4 The Experiment

Due to the fact that these experiments took place in a laboratory setting rather than in the field, participants were asked to pretend that they were recently hired employees of a company who uses SmartRooms to record their meetings, and that their supervisor has asked them to do some fact-finding using the Archivus system. The participants were told in general terms what types of data were available to them. Each participant had to answer a series of true/false (*The budget was 1000CHf?)* and short-answer questions (*Who attended all of the meetings?)*, of which there were 21 in total, conceived in such as way as to encourage the user to explore the different areas and media in the system. The system was designed in such a way that in almost all cases there is more than one way to find a piece of information, and only a small number of questions (3 out of 21) tightly restricted the user as to the media which they had to use to find the answer. As a result, we believe that the users found the most convenient or natural way for them to find the solutions, without being forced to use specific media. Moreover, the order in which the questions were given to the participants was varied in order to reduce any biasing effects that might be introduced by the order of the questions and/or their nature. The 21 questions were divided into 4 sets (2 each of 5 true-false and short-answer questions, plus 1 document-finding question), where the questions within each set were held constant but their order was varied. We also ensured that the order of the sets themselves varied across the different conditions.

The experiment unfolded as follows. First, the participant was given a pre-experiment questionnaire that asked them demographic information, and they were asked to sign a consent form. Then, they were given a description of the scenario and the Archivus software manual so that they could familiarize themselves with the system. They were not allowed to 'play' with the system itself at this time. The experiment proper was done in two phases, each 20 minutes long. In the first phase, the participants were only given access to a subset of all available modalities and had to answer 11 questions (5 true/false, 6 short answer). In the second phase, they were given access to all of the modalities and had to answer 10 questions (5 true/false, 5 short answer). Finally, the participants were given a post-experiment questionnaire followed by a brief interview. The participant was alone in the room and the experimenter only intervened between the two phases to provide the new set of questions.

The results described in the following section are from an experiment involving 24 participants (11 female, 13 male) who were mostly non-native English speakers with average computer experience. A total of 4 modalities were used - mouse (M), voice (V), keyboard (K) and touchscreen (T). These were divided into 8 Phase 1 conditions (M, T, V, M-K, V-K, T-V-K, M-V-K, M-T-V-K, this last group also acting as a control group). The experiment was conducted across subjects, with three subjects assigned per condition. As the low number of participants per condition does not allow for an informative statistical analysis of the data, the results presented in the following section are essentially qualitative.

5 Results and Discussion

In our analysis of the experimental results, we looked at three main aspects – the learning effect, task completion, and the number of interactions.

5.1 Learning Effect

As mentioned in section 4, the experiment was performed in two phases. In Phase 1, users were given limited access to the modalities. This was done in order to investigate whether a modality-learning effect occurred. For example, we expected that participants who only had language input (and in particular voice) available in the first phase would be more inclined to again use language input in the second phase. We based this assumption on the fact that the language-only conditions would give those users more experience with the novel modality than those participants who had limited (via keyboard) or no language input. To our surprise, we found that the opposite was true. Table 1, shows the percentage of use of both language and pointing modalities in Phase 2 for several different Phase 1 conditions.

Those participants who were given the more traditional condition of mouse-keyboard in Phase 1 used language the most in Phase 2. In particular, in this condition the use of voice accounted for over a quarter of all interactions, including mouse and touchscreen interaction. Similarly, participants who had access to all of the modalities in Phase 1 also used language input much more frequently than others in Phase 2. The learning effect that we expected seemed in fact to be a negative effect, in that those participants who had a language-only (voice or voice-keyboard) condition in Phase 1 used language relatively infrequently in Phase 2.

Table 1. Number of interactions in Phase 2[2]

Phase 1 condition	Phase 2 results	
	Mouse/Touchscreen	Voice/Keyboard
Voice only	91%	9%
Voice-Keyboard	88%	12%
Mouse-Keyboard	66%	34%
All modalities	78.5%	21.5%

The lack of a demonstrable learning effect between the two phases leads us to suspect that users have an unconscious need to feel comfortable with the system and with the input modalities that they have available to them at the early stages of interaction. This *comfort* can manifest itself in two ways. The first is in terms of being comfortable with the system itself – knowing what the graphics represent, what type of information is available in the system and where it can be found. This type of comfort was the same for all users in this experiment since none of them had been exposed to the system or the domain before. The second is at the level of interaction with the system, and specifically with which input modalities the user has available to them. This type of comfort was different among the different users' conditions.

Those users who had the mouse-keyboard condition in Phase 1 learned how to interact with the system using a traditional paradigm with which they were already comfortable (mouse-keyboard interaction with a desktop system). Those who had all modalities in Phase 1 also had the option, should they choose to exercise it, to slip into the traditional paradigm. Since it was the users under these two Phase 1 conditions in particular who were more inclined to use language in Phase 2, this suggests that users were more willing to explore using language when they knew that they could fall back on traditional or 'comfort' modalities' (mouse and keyboard) if they needed to.

Those users who had only language input in Phase 1 may have found themselves lost or frustrated during initial interaction with the system since they were faced with a new tool and little (keyboard only) or no familiar modalities on which to rely. Consequently, when given the option to use all modalities in Phase 2, they immediately reverted to the comfortable modality(ies) that they were unconsciously longing for in Phase 1, namely the mouse and/or keyboard. Another possible explanation for this behaviour however could be that in general the response times to language based input were slower and more error prone since the information had to be processed by the wizard. As a result, when users discovered that pointing interaction was faster and more accurate, they would prefer it for reasons of efficiency. Interestingly though, when we looked at the amount of voice use in Phase 2 of the voice-only and mouse-only conditions, we found that in fact, there had been more use of voice input in Phase 2 from the participants who had mouse-only input in Phase 1.

These results lead us to believe that users *are* willing to use language input in general, and voice input in particular, and that the increase in use in Phase 2 is evidence that they see a benefit in doing so (even if it is only at the unconscious level).

[2] All figures in the tables are presented as the mean over the 3 users in each condition.

However, the way in which the use of language is introduced plays an important role in this willingness. Language input only appears to be beneficial if is introduced at a later stage of experience with the system, or ideally, if it is introduced in a mutually exchangeable combination with other more familiar input modalities such as the mouse and keyboard, but in particular the mouse.

5.2 Task Completion

Given the widespread use of the mouse and keyboard and the fairly short amount of time with which users had to familiarize themselves with the system, we expected that the mouse-keyboard combination would be the most effective in finding answers. In the analysis presented in this section, we make a distinction between finding the correct answer and finding an answer, independent of correctness. In the context of the scenario of use for the Archivus system, the user should be able to find the information that they are looking for and have some certainty that it is the correct information. However, in the strict sense of examining which modalities are used when interacting with the system, the distinction between correct and incorrect answers is not important since it is the overall interaction patterns, that we are interested in.

In this section we only consider Phase 1 of the experiment where the user has access to a restricted set of input modalities. Table 2 shows the success rates for finding an answer irrespective of correctness, for each condition in Phase 1 of the experiment.

Table 2. All answers found (independent of correctness) in Phase 1

Phase 1 condition	**Suc.**	**Phase 1 condition**	**Suc.**
Mouse only	40%	Touch -Voice – Keyboard	**26%**
All modalities	36%	Voice only	**26%**
Touchscreen only	33%	Mouse –Keyboard	**26%**
Mouse-Voice-Keyboard	**26%**	Voice – Keyboard	10%

As the data in the table clearly show, our initial expectation that the mouse-keyboard condition would have the highest success rate was not valid. The mouse-keyboard condition is in fact on par with the completely novel modality of voice-only input, and even with the somewhat less novel touchscreen-voice-keyboard and mouse-voice-keyboard conditions. Moreover, we can see that the mouse-only, touchscreen-only and all modalities conditions are the most effective while the voice-keyboard condition is the least effective. An identical ordering of modality conditions can be seen in Table 3 which shows the success rate for finding *correct* answers. This case however, provides a clearer division of success rates across the different conditions.

Tables 2 and 3 suggest that combining voice input with other modalities does add value to the interaction, as evidenced by the low position of the mouse-keyboard condition. Moreover, although mouse-only had the highest success rate it is interesting to note that the all modalities combination was the second most successful. In this condition, language based input (keyboard and/or voice) represented 17% of all of the interactions (not shown in the table). The high success rate of the mouse-only and touchscreen-only conditions can be attributed to the fact that users in these conditions

are strongly constrained in the types of moves (or interactions) they can make. In fact they cannot make any moves that are inappropriate or unknown to the system. This is not the case when voice or keyboard interaction is involved, since the user cannot know ahead of time which terms are known to the system and which are not.

Table 3. Correct answers found in Phase 1

Phase 1 condition	Suc.	Phase 1 condition	Suc.
Mouse only	36%	Touch -Voice-Keyboard	21%
All modalities	25%	Voice only	18%
Touchscreen only	24 %	Mouse-Keyboard	12%
Mouse-Voice-Keyboard	24%	Voice-Keyboard	6%

Finally, participants in the touchscreen-only condition did quite a bit worse than their mouse-only counterparts. In the Archivus system, touchscreen and mouse input are functionally equivalent. Consequently, these results could be attributed to either the lower accuracy rate of using a touchscreen as compared to a mouse when selecting items on the screen or to the fact that users are less familiar with touch as an input modality and are thus slower at using it. Although the difference is less marked, a similar trend can be seen when we compare adding the mouse or the touchscreen (which are functionally equivalent) with the keyboard-voice input.

5.3 Number of Interactions

The final aspect of the data that we looked at was the number of interactions that users made per modality. Due to the fact that language input is inherently slower in the system than pointing input due to linguistic processing time, we could only reasonably compare modalities that are functionally equivalent at the system level (language-only and pointing-only) as we could not determine a time/modality estimate from our data. Table 4 presents the number of interactions made in each phase per modality combination. The difference in the number of interactions between the two phases is given as a percentage in the *Diff* column in the table.

Table 4. Modality interactions per condition and phase

Phase 1 condition	Ph. 1	Ph. 2	Diff.
Mouse only	268	199	- 26 %
Touchscreen only	195	138	- 29%
Mouse-Keyboard-Voice	158	161	2%
Touchscreen-Keyboard-Voice	81	108	25%
Voice only	62	167	63%
Voice-Keyboard	49	124	61%

If we compare the number of interactions between mouse-only vs. touchscreen-only we see that users were more active with the mouse. Given their functional

equivalence we can attribute this to the comfort factor as discussed previously. A similar trend can be seen for mouse-keyboard-voice and touchscreen-keyboard-voice conditions. This could also imply that there is a blocking effect caused by the novelty of the touchscreen. Because users are unfamiliar with the use of a touchscreen, they are more hesitant to use it, thus resulting in a lower number of interactions. Apparently, the novelty is enough to prevent the user from realizing that the two input modalities are functionally equivalent.

The discrepancy between voice-only and voice-keyboard conditions given in the last two rows of Table 4 is likely due to the same effect. Participants using voice-only made more interactions than their functionally equivalent voice-keyboard counterparts. This is interesting since we would assume the opposite to be true as users are already familiar with the keyboard. This could be a case of a novel modality on its own being easier to learn and use than a novel modality in combination with the less frequently used half (keyboard) of the more traditional paradigm of mouse-keyboard discussed in the previous section.

In Table 5 we show the percent of all interactions for voice and keyboard in both phases of the experiment with voice-keyboard as the Phase 1 condition. This suggests a strong link between the modalities of the traditional paradigm.

Table 5. Voice vs. keyboard interactions in Phase 1 and 2

	Phase 1	**Phase 2**
Voice	97%	25%
Keyboard	3%	75%

Voice is used much more frequently than the keyboard in Phase 1, but in Phase 2 where the user also has access to the mouse and the touchscreen, they suddenly begin to use the keyboard much more than voice. We attribute this phenomenon to interference from the comfort that the traditional mouse-keyboard paradigm presents. The user has reverted to using the mouse as input because that is the modality with which they are most familiar. Although they had reasonable success with using voice in Phase 1 and have more experience with it than with keyboard input (in the Archivus system) they still go back to using the keyboard because this is the language input modality that they strongly associate with mouse use. However, there is evidence that this fairly strong interaction paradigm can be weakened under certain circumstances. For example, in Phase 2 of the mouse-keyboard condition, voice was used almost twice as much as the keyboard, despite the continued high use of mouse input.

6 Conclusions

The discussion in the previous section has shown that users *can* be encouraged to use novel input modalities and that these novel input modalities, and natural language input through voice in particular, *do* bring an added benefit to interactions for this domain, at least where the Archivus system is concerned. This is particularly true if users are allowed to use novel modalities together with more familiar modalities such

as the mouse and keyboard. Moreover, while there was evidence of a blocking effect introduced by novel modalities, the all-modalities combination seems to reduce it to a level where it does not significantly interfere with using the system. Finally, we found that even though the desire to interact using the traditional paradigm of mouse-keyboard seemed to be quite strong, it could be weakened in some cases.

These results were achieved in spite of a high learning curve in how to use the software (evidenced by the relatively poor task completion results) and the relatively small number of participants per condition. Both of these problems will be addressed in future work. The data presented here suggest tendencies. In order to show more concrete results, a larger data set and a more in-depth analysis which was not possible with the current data set are needed.

7 Future Work

Encouraged by the preliminary results described in this paper, we have modified the Archivus software to rectify some of the causes of the steep learning curve discovered with the previous version. Additionally, we have replaced the touchscreen with a stylus-driven tablet PC. This was done because we believe that the tablet PC is a more realistic platform for this type of software, and also because it provides greater accuracy in selection of items in the interface than the touchscreen did, which should strengthen the comparison between physically pointing and using the mouse as input modalities. Evidence gathered from pilot experiments suggests that the new version of the software is easier to learn and use, and there seems to be an increase in the overall use of the different modalities. The next step will be to run a large-scale experiment with Archivus to examine multimodal interaction with the new input modalities (mouse, voice, keyboard and stylus on the tablet PC) in the summer of 2006.

Acknowledgements

The authors would like to thank Martin Rajman, Mirek Melichar and Marita Ailomaa for their collaboration on the design, development and evaluation of the Archivus system, and the SNSF and the University of Geneva for funding this research.

References

1. Bernsen, N.O. and Dybkjaer, L.: Evaluation of Spoken Multimodal Conversation. In proceedings of the International Conference on Multimodal Interfaces, ICMI -04. (October 14-15, 2004 State College, Pennsylvania, USA), ACM Press, pp. 38-45
2. Dahlbäck, N., Jönsson, A., Ahrenberg, L.: Wizard of Oz Studies - Why and How. In International Workshop on Intelligent User Interfaces. 1993. Orlando, FL, USA: ACM Press, New York.
3. Karat, C., Vergo, J., and Namahoo D.: Converstional Interface Technologies. In The Human-Computer Interaction Handbook. Jacko, J.A., and Sears, A. (eds). Lawrence Erlbaum Associates Inc., Mahaw, New Jersey. 2003, pp 169-187.

4. Lai, J., and Yankelovich, N.: Conversational Speech Interfaces. In The Human-Computer Interaction Handbook. Jacko, J.A., and Sears, A. (eds). Lawrence Erlbaum Associates Inc., Mahaw, New Jersey. 2003, pp 698-714.
5. Lisowska, A., Popescu-Belis, A., Armstrong, S.: User Query Analysis for the Specification and Evaluation of a Dialogue Processing and Retrieval System. In LREC'2004 (Fourth International Conference on Language Resources and Evalu Tion). 2004. Lisbon, Portugal.
6. Lisowska, A., Rajman, M., Bui, T.H.: ARCHIVUS : A System for Accesssing the Content of Recorded Multimodal Meetings. In MLMI - Joint AMI/PASCAL/IM2/M4 Workshop on Multimodal Interaction and Related Machine Learning Algorithms. 2004. Martigny, Switzerland.
7. Moore, D.: The IDIAP Smart Meeting Room. IDIAP: Martigny (Switzerland), 2002. p. 13.
8. Oviatt, S.: Multimodal Interfaces, In The Human-Computer Interaction Handbook. Jacko, J.A., and Sears, A. (eds). Lawrence Erlbaum Associates Inc, Mahaw, New Jersey, 2003, pp 286-304.
9. Salber, D., Coutaz, J.: Applying the Wizard of Oz Technique to the Study of Multimodal Systems. In 3rd International East/West Human Computer Interaction Conference. 1993. Moscow, Russia: Springer Verlag Publ.
10. Tucker, S., Whittaker, S.: Accessing Multimodal Meeting Data: Systems, Problems and Possibilities. In, S. Bengio and H. Bourlard, (eds): Lecture Notes in Computer Science 3361. Springer Verlag, 2005. p. 1-11.

Gesture Features for Coreference Resolution

Jacob Eisenstein and Randall Davis

Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology, Cambridge MA 02139, USA
{jacobe,davis}@csail.mit.edu

Abstract. If gesture communicates semantics, as argued by many psychologists, then it should be relevant to bridging the gap between syntax and semantics in natural language processing. One benchmark problem for computational semantics is *coreference resolution*: determining whether two noun phrases refer to the same semantic entity. Focusing on coreference allows us to conduct a quantitative analysis of the relationship between gesture and semantics, without having to explicitly formalize semantics through an ontology. We introduce a new, small-scale video corpus of spontaneous spoken-language dialogues, from which we have used computer vision to automatically derive a set of gesture features. The relevance of these features to coreference resolution is then discussed. An analysis of the timing of these features also enables us to present new findings on gesture-speech synchronization.

1 Introduction

Although the natural-language processing community has traditionally focused mainly on text, the actual usage of natural language between people is primarily oral and face-to-face. Extension of robust NLP to face-to-face communication offers the potential for breakthrough applications in domains such as meetings, lectures, and presentations. We believe that in face-to-face discourse, it is important to consider the possibility that non-verbal communication may offer information that is crucial to language understanding. However, due to the long-standing emphasis on text datasets, there has been little work on non-textual features.

In this paper, we investigate the relationship between gesture and semantics. We use machine vision to extract hand positions from a corpus of sixteen videos. We present a set of features that are derived from these hand positions, and use statistical methods to characterize the relationship between the gesture features and the linguistic semantics. Semantics is captured concretely in the context of *coreference*, which occurs when two noun phrases refer to the same entity. If gesture features can predict whether two noun phrases corefer, then they can contribute to the semantic analysis of speech.

2 Corpus

To conduct this research, we have begun to gather a corpus of multimodal dialogues. This work is preliminary, and the size of the corpus is relatively small; as we will describe in more detail at the end of Section 3, our corpus is roughly half the size of

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 154–165, 2006.

the MUC-6 coreference evaluation formal corpus [1]. We hope that interest in multimodal natural language processing will increase, leading to the development of better and broader corpora.

2.1 Procedure

Thirty college students and staff, aged 18-32, joined the study by responding to posters on our university campus. A subset of nine pairs of participants was selected on the basis of recording quality, and their speech was transcribed and annotated. The corpus is composed of two videos from each of the nine pairs; technical problems forced us to exclude two videos, yielding 16 annotated documents, each between two and three minutes in duration.

McNeill [2] and others have long advocated studying dialogues in which the speaker and listener already know each other. This eliminates a known confound in which the speaker and listener increase the rate of gestures as they become acquainted over the course of the experiment. For this reason, we recruited participants in pairs; 78% of participants described themselves as "close friends" or spouses; 20% as "friends", and 3% as "acquaintances".

One participant was randomly selected from each pair to be the "speaker," and the other to be the "listener." The speaker's role was to explain the behavior of a mechanical device to the listener. The listener's role was to understand the speaker's explanations well enough to take a quiz given later. The listener was allowed to ask questions of the speaker; however the listener's speech has not yet been transcribed, and is not considered in this study.

Prior to each discussion, the speaker either privately viewed a simulation of the device in operation, or left the room and examined the actual physical object. In explaining the device, the speaker was provided with either a whiteboard marker with which to create a sketch, a pre-printed diagram of the device, or no visual aids at all. In this paper, we will consider only data from the condition with the pre-printed visual aid. The interpretation of gestures in this condition is thought to be more straightforward; many, if not most of the gestures involve pointing at locations on the diagram. While the (presumably) more challenging problem of understanding gesture without visual aids is interesting future work, printed or projected diagrams are common in business presentations and lectures, so this restriction does not seem to be overly limiting to the applicability of our work.

The speaker was limited to two minutes to view the video or object and three minutes to explain it; the majority of speakers used all of the time allotted. This suggests that we could have obtained more natural data by not limiting the explanation time. However, we found in pilot studies that this led to problematic ordering effects, where participants devoted a long time to the early conditions, and then rushed through later conditions. With these time constraints, the total running time of the experiment was usually around 45 minutes. More details on the data-gathering portion of this research can be found in [3].

2.2 Speech and Vision Analysis

Speech was recorded using headset microphones. A homebrew Java system controlled the synchronization of the microphones and video cameras. Speech was transcribed

manually, and audio was hand-segmented into well-separated chunks with duration not longer than twenty seconds. The chunks were then force-aligned by the SPHINX-II speech recognition system [4].

Video recording was performed using standard digital camcorders. Participants were given different colored gloves to facilitate hand tracking. Despite the use of gloves, a post-study questionnaire indicated that only one of the thirty participants guessed that the study was related to gesture. The study was deliberately designed so that participants had very little free time to think; when not actually conducting the dialogue, the speaker was busy viewing the next mechanical system, and the participant was busy being tested on the previous conversation. We also presented consent forms immediately after the gloves, which may have diverted attention from the gloves' purpose.

An articulated upper-body tracker was used to model the position of the speaker's torso, arms, and hands. By building a complete upper-body tracker, rather than simply tracking the individual hands, we were able to model occlusion directly. Search across configurations of the tracker was performed using an annealed particle filter, implemented using the OpenCV library.[1] Essentially, the system performed a randomized beam search to simultaneously achieve three objectives: a) maximize coverage of the foreground area, b) match the known glove color to the color observed at the hypothesized hand positions, c) respect physiological constraints and temporal continuity.

The tracker was based largely on the work of [5]; the main differences were that Deutscher et al. did not use color cues such as gloves, but had access to multiple cameras to facilitate 3D tracking. We used only a single monocular camera and a 2.5D model (with just one degree of freedom in the depth plane). From inspection, the lack of depth information was the cause of many of our system's errors; bending of the arm joints in the depth dimension caused the arm length to appear to change in ways that were confusing to our model. Nonetheless, we estimate that both hands were tracked accurately and smoothly over 90% of the time. It is difficult to assess the tracker performance more precisely, as that would require ground truth data in which the actual hand positions were annotated manually at each time step.

3 Annotation

Coreference annotation is a two-step process [6]. First, all noun phrases (NPs) that may participate in coreference relations are selected; these are sometimes called "markables." All third-person noun phrases were included as markables; in addition, nested markables were annotated (e.g., "[one of [these walls]]" [2]). First and second person NPs were excluded as markables, as they were thought to be irrelevant to the issue of understanding the explanatory narratives about physical devices; furthermore, it seemed unlikely that gesture could play much of a role in disambiguating coreferences among such entities. It would be easy to automatically separate out these NPs in most cases. Manual annotation also disambiguated different senses of words like "that," which can be a referential pronoun (e.g. "that is rotating") or a relative pronoun "the one that rotates." The word "there" and "it" were also not included as markables when they did not indicate entities, as in "there's no way out of here," and "it's hard to tell."

[1] http://www.intel.com/technology/computing/opencv/

[2] In this notation, noun phrases are set off by brackets.

The corpus consists of spontaneous speech, so disfluencies abound. Repeated word disfluencies were automatically eliminated when the repetitions were adjacent, but other disfluencies were left uncorrected; rather than performing coreference resolution on an error-free text, we include the markables that occur inside disfluencies without prejudice. A frequent type of disfluency involves restatement of a noun phrase, e.g., "so this pushes [these] [all these things] up." This is a *substitution* disfluency, where "all these things" substitutes for "these." However, both are treated as markables, with a coreference relation between them, since they refer to the same set of objects.

A total of 1141 markables were found in the corpus, an average of 71 per video sequence ($\sigma = 27$). After the markables were annotated, the second step is to specify the coreference relations between them. As with the selection of markables, coreference annotation was performed by the first author, following the MUC-7 task definition [6]. A coreference relation was annotated whenever two noun phrases were judged to have an identical reference. 74.5% of markables participated in coreference relations, and there were a total of 474 entities, yielding a markable-to-entity ratio of 2.4.

4 Features

To assess the relationship between gestures and coreference, we computed a set of features describing the position and motion of the tracked hands. Two of the these features are *comparative*, in that they can be used to measure the similarity of gestures during different points in time. These can be applied directly to coreference, comparing the gestures observed during the two candidate noun phrases. Five other features are not comparative, meaning that they describe only a single gesture. These features can be used to assess the likelihood that an individual noun phrase relates to *any* previously defined entity, or the likelihood that the comparative features will be applicable to determining coreference.

Some of the features invoke the idea of "focus": which hand, if any, is gesturing during the utterance of the noun phrase. There are four logical possibilities: both, left, right, or neither. For the moment we ignore bimanual gestures, which were not frequent in our data. The determination of which hand is in focus is governed by the following heuristic: select the hand farthest from the body in the x-dimension, as long as the hand is not occluded and its y-position is not below the midsection of the speaker's body. If neither hand meets these criteria, than no hand is said to be in focus.

In the notation that follows, $x_{start_j,L}$ refers to the x-position of the left hand at the start of the noun phrase j. For the non-comparative features, there is only one noun phrase, and so no need to index them. $y_{start,F}$ refers to the y-position of whichever hand is in focus at the start of the noun phrase.

4.1 Comparative Features

– FOCUS DISTANCE: The distance between the positions of the in-focus hand during the two candidate noun phrases. The focus distance is undefined if there is no focus hand during either candidate noun phrase. FOCUS DISTANCE is given by

$$\sqrt{(x_{midpoint_j,F_j} - x_{midpoint_i,F_i})^2 + (y_{midpoint_j,F_j} - y_{midpoint_i,F_i})^2}. \quad (1)$$

– WHICH HAND: Takes three values: *SAME*, if the same hand is in focus during both candidate NPs; *DIFFERENT*, if a different hand is in focus; *MISSING*, if no focus hand is found during at least one of the NPs.

4.2 Non-comparative Features

– FOCUS SPEED: The total displacement of the in-focus hand, divided by the duration of the word. FOCUS SPEED is undefined if there is no focus hand at either the beginning or end of the word, or if the focus hand is different at the end of the word from the focus hand at the beginning. Note that by this metric, the FOCUS SPEED is zero if a gesture ends in the same place that it begins, regardless of how much distance the hand traversed. FOCUS SPEED is given by

$$\sqrt{(x_{end,F} - x_{start,F})^2 + (y_{end,F} - y_{start,F})^2}/\text{duration}. \tag{2}$$

– JITTER: JITTER measures the average frame-by-frame speed of each hand, over the course of the NP. Since this feature does not require the determination of the focus hand, it is never undefined. However, contributions at instants when a hand is occluded are not counted in the sum. This feature was found to be more informative when smoothed by a Gaussian kernel. JITTER is given by

$$\sum_{t \geq start}^{end} \frac{\sqrt{(x_{t,L} - \bar{x}_L)^2 + (x_{t,R} - \bar{x}_R)^2 + (y_{t,L} - \bar{y}_L)^2 + (y_{t,R} - \bar{y}_R)^2}}{\text{duration}}. \tag{3}$$

– PURPOSE: This feature captures how much of the overall motion (the JITTER) is explained by purposeful motion between the start and end points of the gesture. PURPOSE is simply FOCUS SPEED / JITTER; it is defined to be zero whenever jitter is zero, and is undefined whenever FOCUS SPEED is undefined. The value is maximized for fast, directed motions that go linearly between the start point and the end point; it is minimized for erratic motions for which the end point is not far from the start point. As with FOCUS SPEED, this feature will give a low score to any periodic motion, even large circles.
– SYNCHRONIZATION: SYNCHRONIZATION measures the degree to which the two hands are moving on the same trajectory. Like JITTER, this feature does not use focus information, and is therefore never undefined. However, contributions at instants when a hand is occluded are not counted in the sum. The value of this feature is 1 if the two hands are perfectly synchronized, 0 if the hands are moving orthogonally, and -1 if the hands are antisynchronized (i.e., one hand is moving clockwise and the other counterclockwise). $atan2$ is the full circle arctan function, which returns values in the range $\{-\pi, \pi\}$.

$$\theta_{t,h} = atan2(y_{t,h} - y_{t-1,h}, x_{t,h} - x_{t-1,h}) \tag{4}$$

$$\text{synch}_t = cos(\theta_{t,L} - \theta_{t,R}) \tag{5}$$

$$\text{synch} = \sum_{t > start}^{end} \frac{\text{synch}_t}{\text{duration}} \tag{6}$$

– SCALED SYNCHRONIZATION: SCALED SYNCHRONIZATION scales the SYNCHRONIZATION score at each time step by the average of the speeds of the two hands at that time step. Thus, large synchronized movements are weighed more heavily than small ones. The velocities $v_{t,h}$ – referring to the velocity of hand h at time t – are smoothed using a Gaussian kernel.

$$v_{t,h} = \sqrt{(x_{t,h} - x_{t-1,h})^2 + (y_{t,h} - y_{t-1,h})^2} \tag{7}$$

$$\text{scaled\_synch} = \frac{\sum_{t>start}^{end}(v_{t,L} + v_{t,R})\text{synch}_t}{2 * \text{duration}} \tag{8}$$

5 Feature Relevance

Both of the comparative features were predictive of coreference. FOCUS DISTANCE, which measures the distance between the hand positions during the two noun phrases, varied significantly depending on whether the noun phrases coreferred (see Table 1). The average noun phrase has a FOCUS DISTANCE of 48.4 pixels to NPs with which it corefers ($\sigma = 32.4$), versus a distance of 74.8 to NPs that do not corefer ($\sigma = 27.1$); this difference is statistically significant ($p < .01, dof = 734$). We expected to find larger differences for pronouns or NPs starting with the word "this," expecting gesture to play a larger role is disambiguating such NPs. As shown in the table, this does not appear to be the case; there was no significant increase in the discriminability of the FOCUS DISTANCE feature for such linguistic phenomena. However, FOCUS DISTANCE does appear to play less of a discriminative role for definite NPs (e.g., "the ball") than for non-definite NPs ($t = 3.23, dof = 183, p < .01$). This suggests that the definite article is not used as frequently in combination with gestures that communicate meaning by hand location, and that computer systems may benefit by attending more to gesture during non-definite NPs.

Table 1. Average values for the FOCUS DISTANCE feature (Equation 1), computed for various linguistic phenomena

	coreferring	not coreferring	difference
all	48.4	74.8	26.4
pronouns	50.3	76.5	26.2
non-pronouns	46.5	73.5	27.0
definite NPs	50.8	70.0	19.2
non-definite NPs	48.0	75.7	27.7
"this"	44.5	71.8	27.3
non-"this"	50.2	76.1	25.9

The FOCUS DISTANCE feature is based on a Euclidean distance metric, but this need not be the case; it is possible that coreference is more sensitive to movement in either the y- or x- dimension. Figure 1 helps us to explore this phenomenon: it is a contour plot comparing relative hand position (indicated by the position on the graph) to the

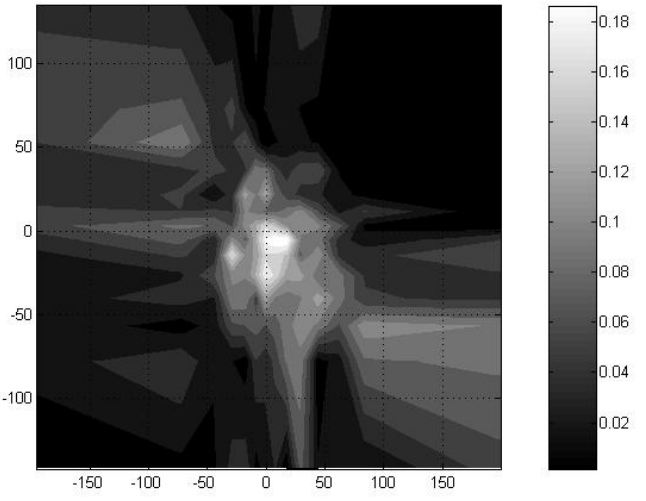


Fig. 1. A contour plot of coreference likelihood based on the relative position of the focus hands. The level of brightness indicates the likelihood of coreference, quantified in the color bar on the right.

likelihood of coreference (indicated by brightness). At (0,0), the hand positions during the candidate NPs are identical; at (0,50), the x-coordinates are identical but the y-coordinates of the focus hand at the time of the anaphoric noun phrase is 50 units higher than at the time of the antecedent NP.

The graph shows that the likelihood of coreference is greatest when the hand positions are identical (at the origin), and drops off nearly monotonically as one moves away from the center. However, there are some distortions that may be important. The drop-off in coreference likelihood appears to be less rapid in the y-dimension, suggesting the accuracy in the y-coordinate is less important than in the x-coordinate. This may be explained by the fact that the figures we used were taller than they were wide – the exact dimensions were 22 by 17 inches – or it may be a more general phenomenon. The decrease in coreference likelihood also appears to be less rapid as one moves diagonally from upper-left to lower-right. This may be an artifact of the fact that speakers more often stood to the left of the diagram, from the camera's perspective. From this position, it is easier to move the hand along this diagonal than from the upper-right to lower-left corners.

The choice of gesturing hand was also found to be related to coreference. As shown in Table 2, speakers were more likely to use the same hand in both gestures if the associated noun phrases were coreferent. They were less likely to use different hands, and they were more likely to gesture overall. These differences were found to be statistically significant ($p < .01, \chi^2 = 57.3, dof = 2$). Table 2 also shows how the relationship between hand choice and coreference varied according to the type of noun phrase. Hand choice was a significant feature for all types of NPs, except those beginning with the word "this" ($p = .12, \chi^2 = 4.28, dof = 2$).

Table 2. Hand choice and coreference. All figures are percentages.

	same	*different*	*no gesture*
overall			
coreferring	59.9	19.9	20.2
not coreferring	52.8	22.2	25.1
all	53.2	22.0	24.8
anaphor is a pronoun			
coreferring	58.9	18.6	22.6
not coreferring	50.5	21.0	28.5
all	51.2	20.8	28.0
anaphor is definite NP			
coreferring	57.8	25.6	16.6
not coreferring	54.7	22.2	23.1
all	54.9	22.5	22.7
anaphor begins with "this"			
coreferring	65.6	21.8	12.6
not coreferring	61.6	23.0	15.4
all	61.8	22.9	15.3

The overall rate of detected gestures varied across linguistic phenomena, shown in column 4. The fewest gestures occurring during pronouns, and the most occurring during NPs beginning with the word "this." It is somewhat unsettling that the greatest proportion of gestures occurred during such NPs, and yet our comparative features were not better at predicting coreference – in fact, the WHICH HAND feature was worse for these NPs than overall.

5.1 Non-comparative Features

Non-comparative features cannot directly measure whether two noun phrases are likely to corefer, as they can be applied only to a single instant in time. However, they can be used for at least two purposes: to determine whether a given noun phrase is likely to have *any* coreferent NPs, and to assess whether the comparative features will be useful. In other words, non-comparative features may tell us whether gesture is worth attending to.

The first question – the relationship between non-comparative features and anaphoricity – is addressed in Table 3. For a given noun phrase, its "children" are subsequent NPs that corefer to it; its "parents" are preceding NPs to which it corefers. Columns two and three list the mean feature values, conditioned on whether the NP has children. Columns five and six are conditioned on whether the NP has parents.

As indicated by the table, the FOCUS SPEED and PURPOSE features predict the presence of "children" NPs when they take on low values. Both features attempt to quantify whether the hand is being held in position during the associated NP, or if it is in the process of moving to another location. The presence of a hold during a noun phrase seems to predict that future NPs will refer back to the present noun phrase; perhaps

Table 3. Non-comparative features, with 95% confidence intervals

feature	children	no children	p	parents	no parents	p
FOCUS SPEED	$.0553 \pm .0056$	$.0688 \pm .010$	.05	$.0634 \pm .0068$	$.0557 \pm .0079$	
JITTER	$1.37 \pm .066$	$1.36 \pm .079$		$1.39 \pm .067$	$1.34 \pm .079$	
PURPOSE	$.0561 \pm .0066$	$.0866 \pm .020$	.01	$.0656 \pm .0093$	$.0705 \pm .017$	
SYNCHRONIZATION	$.0273 \pm .043$	$.0515 \pm .049$		$.0281 \pm .046$	$.0504 \pm .045$	
SCALED SYNCH	$.095 \pm .096$	$.085 \pm .12$		$.0969 \pm .11$	$.0826 \pm .10$	

speakers are more likely to produce gestural holds when describing concepts that they know are important.

In contrast, none of the features were capable of predicting whether a noun phrase had parents – previously uttered NPs to which it refers. This is somewhat disappointing, as we had hoped to find gesture cues indicating whether an entity was being referred to for the first time. As always, it is possible that this information is carried in gesture, and our feature set is simply insufficient to capture it.

Non-comparative features as meta-features. Finally, we consider the possibility that non-comparative features can serve as meta-features, telling us when to consider the comparative features. If so, we would expect to observe a non-zero correlation between useful meta-features and the discriminability of the comparative features. That is, the meta feature is helpful if for some values, it can alert us that the comparative feature is likely to be highly discriminable.

For a given noun phrase NP_j, the discriminability of the FOCUS DISTANCE feature can be measured by subtracting the average FOCUS DISTANCE to all coreferring noun phrases NP_i from the average FOCUS DISTANCE to all non-coreferring noun phrases:

$$
\begin{aligned}
cr(NP_j) &= \text{Set of all noun phrases coreferent with } NP_j\text{, not including } NP_j \text{ itself.} \\
\overline{cr}(NP_j) &= \text{Complement of } cr(NP_j)\text{, not including } NP_j \text{ itself.} \\
\text{FD}(NP_j, NP_i) &= \text{FOCUS DISTANCE between } NP_j \text{ and } NP_i\text{, as defined in Equation 1}
\end{aligned}
$$

$$
d(NP_j) = \frac{1}{|\overline{cr}(NP_j)|} \sum_{\overline{cr}(NP_j)} \text{FD}(NP_j, NP_i) - \frac{1}{|cr(NP_j)|} \sum_{cr(NP_j)} \text{FD}(NP_j, NP_i) \quad (9)
$$

For each NP_j, we can measure the correlation of the non-comparative features at NP_j with the discriminability $d(NP_j)$, to see whether the non-comparative features are indeed predictive of the discriminability of the FOCUS DISTANCE feature. The results are shown in Table 4. Significance values are computed using the Fisher transform.

Table 4 shows that low FOCUS SPEED and PURPOSE are indicative of a possibly useful gesture, which is expected, since they are indicative of a gestural hold. Additionally, the x-distance from the center of the body is also predictive of gesture discriminability. This reflects the fact that useful gestures typically refer to the diagram, and the speakers are usually mindful not to stand in front of it. The last two lines of the table show results for linear and interaction regression models, suggesting that a classifier built using these features could be strongly predictive of whether the current gesture is informative. This

would suggest a meta-learning system that could tell us when to pay attention to gesture and when to ignore it.

6 Gesture-Speech Synchronization

Given that gesture carries semantic content associated with speech, it is logical to ask how the speech and gesture are synchronized. Does the more informative part of gesture typically precede speech, follow it, or synchronize with it precisely? This is relevant from both an engineering and scientific perspective. To build systems that use gesture features, it is important to know the time window over which those features should be computed. From a psycholinguistic perspective, the synchronization of gesture and speech is thought to reveal important clues about how gesture and speech are produced and represented in the mind [7,8].

We compared the effectiveness of the FOCUS DISTANCE feature at assessing coreference identity, using discriminability as defined in Equation 9, evaluated at varying temporal offsets with respect to the start, midpoint, and end of the associated noun phrases. The results are shown in Figure 2. The optimal discriminability is found 108 milliseconds after the midpoint of the associated noun phrase. With respect to the speech onset, optimal discriminability is found 189 milliseconds after the speech onset. With respect to the end of the lexical affiliate, the optimal discriminability is found 81 milliseconds after the end point; while it is somewhat surprising to see the optimal discriminability after the end of the word, this may be the result of noise, as this graph is less sharply peaked than the other two.

Some of the existing psychology literature suggests that gesture strokes typically *precede* the associated speech [7,8]. Our data may appear to conflict with these results, but we argue that in fact they do not. The existing literature typically measures synchronization with respect to the *beginning* of the gesture and the beginning of speech. Note that the FOCUS DISTANCE feature only captures the semantics of deictic gestures, which McNeill defines as gestures that communicate meaning by hand position [2]. In particular, we believe that our results capture the beginning of the post-stroke "hold" phase of the gesture: exactly the moment at which the movement of the gesture ends, and the

Table 4. Non-comparative feature correlations with FOCUS DISTANCE discriminability

feature	r	**significance** (dof = 377)
FOCUS SPEED	-.169	$p < .01$
JITTER	.0261	
PURPOSE	-.1671	$p < .01$
SYNCHRONIZATION	-.0394	
SCALED SYNCHRONIZATION	-.0459	
Y-distance from bottom	.0317	
X-distance from body center	.215	$p < .01$
Regression, linear model	.288	$p < .01$
Regression, interaction model	.420	$p < .01$

hand rests at a semantically meaningful position in space. This analysis is supported by the finding that the FOCUS DISTANCE discriminability is negatively correlated with hand movement speed (see Table 4). At the onset of gesture motion, the hand is not yet at a semantically relevant location in space, and so the discriminability of the FOCUS DISTANCE feature cannot capture when motion begins. In future work, we hope to consider whether segmentation of hand motion into movement phases could be automated, facilitating this analysis.

7 Related Work

The psychology literature contains many close analyses of dialogue that attempt to identify the semantic contribution of gesture (e.g., [2,9]). This work has been instrumental in documenting the ways in which gesture and speech interact, identifying features of gesture (termed "catchments"), and showing how they relate to semantic phenomena. However, much of this analysis has been *post facto*, allowing psychologists to bring to bear human-level common-sense understanding to the interpretation of the gestures in dialogues. In contrast, we focus our analysis on coreference – rather than asking "what does this gesture *mean*?", we ask the simpler, yes/no question, "do these two gestures refer to the same thing?" Thus, we are able to systematize our treatment of semantics without having to create an ontology for the semantics of the domain. This, in combination with automatic extraction of gesture features through computer vision, clears the way for a predictive analysis without human intervention. We believe such an analysis provides a useful complement to the existing work that we have cited.

Another relevant area of research is in gesture generation, which has developed and exploited rich models of the relationships between gesture and semantics (e.g., [10]). However, there is an important difference between generation and recognition. Humans are capable of perhaps an infinite variety of gestural metaphors, but gesture generation need only model a limited subset of these metaphors to produce realistic gestures. To understand the gestures that occur in unconstrained human communication, a more complete understanding of gesture may be necessary.

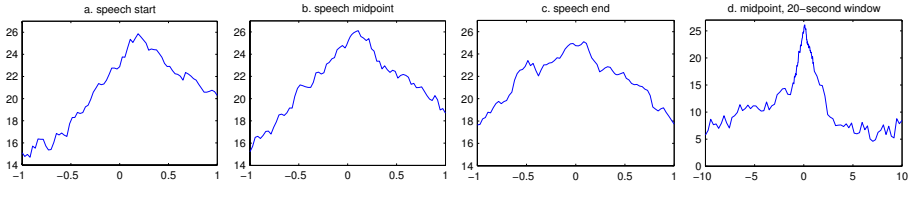


Fig. 2. Gesture-speech synchronization data. The y-axis is the discriminability of the FOCUS DISTANCE feature (Equation 9), and the x-axis is the offset in seconds. Part (d) shows a wider window than the other three plots. Parts (b) and (d) show the discriminability when the offset is computed from the speech midpoint. In parts (a) and (c), the offset is computed from the start and end points respectively.

8 Conclusion

This paper introduces a new computational methodology for addressing the relationship between gesture and semantics, avoiding time-consuming and difficult gesture annotation. We have found several interesting connections between gesture features and coreference, using a new corpus of dialogues involving diagrams of mechanical devices. In our data, the position of gestural holds appears to be the most important gesture feature, and other gesture features can help to determine when holds are occurring. In addition, an analysis of the timing of gesture/speech synchrony suggests that the most useful information for deictic gestures is located roughly 100 milliseconds after the midpoint of the lexical affiliate.

References

1. Grishman, R., Sundheim, B.: Design of the MUC-6 evaluation. In: Proceedings of the 6th Message Understanding Conference. (1995)
2. McNeill, D.: Hand and Mind. The University of Chicago Press (1992)
3. Adler, A., Eisenstein, J., Oltmans, M., Guttentag, L., Davis, R.: Building the design studio of the future. In: Making Pen-Based Interaction Intelligent and Natural, Menlo Park, California, AAAI Press (2004) 1–7
4. Huang, X., Alleva, F., Hwang, M.Y., Rosenfeld, R.: An overview of the Sphinx-II speech recognition system. In: Proceedings of ARPA Human Language Technology Workshop. (1993) 81–86
5. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2000) 126–133
6. Hirschman, L., Chinchor, N.: MUC-7 coreference task definition. In: Message Understanding Conference Proceedings. (1997)
7. Butterworth, B., Beattie, G.: Gesture and silence as indicators of planning in speech. In: Recent Advances in the Psychology of Language, Plenum Press (1978) 347–360
8. Morrel-Samuels, P., Krauss, R.M.: Word familiarity predicts temporal asynchrony of hand gestures and speech. Journal of Experimental Psychology: Learning, Memory and Cognition **18** (1992) 615–623
9. Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X.F., Kirbas, C., McCullough, K.E., Ansari, R.: Multimodal human discourse: gesture and speech. ACM Transactions on Computer-Human Interaction (TOCHI) (2002) 171–193
10. Kopp, S., Tepper, P., Ferriman, K., Cassell, J.: Trading spaces: How humans and humanoids use speech and gesture to give directions. Spatial Cognition and Computation **In preparation** (2006)

Syntactic Chunking Across Different Corpora

Weiqun Xu, Jean Carletta, and Johanna Moore*

HCRC and ICCS, School of Informatics, University of Edinburgh

Abstract. Syntactic chunking has been a well-defined and well-studied task since its introduction in 2000 as the CONLL shared task. Though some efforts have been further spent on chunking performance improvement, the experimental data has been restricted, with few exceptions, to (part of) the Wall Street Journal data, as adopted in the shared task. It remains open how those successful chunking technologies could be extended to other data, which may differ in genre/domain and/or amount of annotation. In this paper we first train chunkers with three classifiers on three different data sets and test on four data sets. We also vary the size of training data systematically to show data requirements for chunkers. It turns out that there is no significant difference between those state-of-the-art classifiers; training on plentiful data from the same corpus (switchboard) yields comparable results to Wall Street Journal chunkers even when the underlying material is spoken; the results from a large amount of unmatched training data can be obtained by using a very modest amount of matched training data.

1 Introduction

Very successful syntactic chunkers have already been developed using part of the Wall Street Journal (WSJ) data from Penn Treebank (PTB) [1] for training and testing in the framework of CONLL 2000 shared task [2]. But so far extensive research on chunking has concentrated solely on making performance gains for this one corpus. The problem of chunking new data, which are usually from other domains/genres, possibly more different from newspaper texts, such as transcribed or even recognised spontaneous speech (e.g., from the AMI corpus [3]), has not received enough attention yet. But for the task of chunking per se, we need to take a step further to widen the scope of chunking technology through chunking new data, in addition to deepening it through improving performance on a specific data set. This also has some practical implications since chunk information has potential for higher-level tasks, like semantic role labelling [4], or some text and speech end applications. Chunking (or partial parsing) should also be able to help when full parsing is very difficult or even impossible.

There are some interesting issues that chunking new data could present. How will those state-of-the-art classifiers work on new data? How will the chunkers trained on old data (WSJ) work on new data (like AMI), and vice versa? Or how

* This work was carried out under funding from the European Commission (AMI, FP6-506811).

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 166–177, 2006.

will the genre and size of data affect chunking performance? In this paper, two series of experiments were carried out to address these issues. First we trained chunkers with three different classifiers on three different corpora, which differ in several aspects, including genre (or similarity) and size, among others. Our goal is not to maximise performance for any of them, but instead to show the effect of corpus and classifier difference on chunking performance. We further ran experiments to show the effect of training data size on chunking performance. Together we would like to inform some strategy for developing chunkers for real applications that use new material, yet within the reach of existing approaches, instead of resorting to advanced techniques like domain adaption or learning from unlabelled data. Therefor our goal remains very modest.

In what follows we first give a brief review of syntactic chunking, then introduce the data and classifiers used. After that, we will present two series of chunking experiments on data of different genre and different size. Finally we conclude with some discussions.

2 Review

Chunking is the task of "dividing text into syntactically related non-overlapping groups of words" [2, p.127]. It can be considered a form of partial parsing, and was partly motivated by the psychological evidence that the human parser works on non-recursive cores of major phrases [5]. Abney both pioneered the idea of parsing by chunks and built a knowledge-intensive, rule-based chunker [6].

More recently, chunking has been reformulated as a task similar to part-of-speech (POS) tagging [7]. In this approach, if only noun chunks are required, then the words from a corpus might be marked up using three tags: B for the initial word of a noun chunk, I for a non-initial word of a noun chunk, and O for a word outside any chunk. Tagging more chunk types requires a larger tag set that has B and I categories for each chunk type; for instance, a noun and verb chunker might use a representation that has the tag set {B-N, I-N, B-V, I-V, O}. This reformulation allows the many learning approaches that have been developed for sequential classification and which are familiar from POS tagging to be applied directly to the problem of chunking [8, 9].

The first community competition for chunking technology was held in CONLL-2000. In the chunking shared task, which used part of WSJ articles as the base material, the organisers created training and test data for not just noun and verb chunks but a wide variety of chunk types by converting some of the syntactic annotation distributed as part of the PTB into tags using the basic reformulation described above.[1] The agreed evaluation metrics for the task were precision, recall, and an F score ($F_{\beta=1}$, which gives precision and recall equal weight). The best performance among the eleven participating systems was F1=93.48 [10]. The main choices that developers of chunkers have exercised are the set of features to use when classifying, exactly how to reformulate the chunks as a tag

[1] The script used for this conversion is available from `http://ilk.kub.nl/~sabine/chunklink`.

set, and which classifier to use. Common features used include the current word being tagged; words in windows of various sizes extending into the left context of the current word, and possibly also into the right context; POS tags for all of these words including the current word; and chunk tags for words in the left (or right, if chunking backward) context windows. Alternative reformulations typically augment the basic dichotomy between chunk-initial (B) and non-initial (I) words with distinctions for the I category that differentiate based on position in the chunk, particularly picking out chunk-final words.

After the CONLL 2000 shared task, various efforts have been spent on improving chunking performance on the common data, e.g., F1 reached 93.91 with SVM voting [11], 93.57 with generalised Winnow [12], 93.71 with filtering-ranking perceptron learning [13], 94.39 with additional unlabelled data [14], etc.

All chunking work to date has focused solely on performance improvements for WSJ texts. However, one paper, [9], has considered the effects of training on data drawn from the SWBD corpus, which contains conversational dialogues, but testing on WSJ data. Osborne carried out this test not because he was interested in the effects of training on one corpus and testing on another, but as a way of estimating the effects of noisy training data. That is, he was using the known degraded grammaticality associated with speech as an approximation for noise in training data. For instance, using section 2 of the SWBD corpus for his training data, a maximum entropy-based classifier, and a feature set that included the last four letters of the current word plus POS labels for the current and two following words, he achieved a performance of F1=82.97. Although he did originally intend his data for these purposes, his results can be reconstructed as more data points about the effects of training on one corpus for testing another from a fairly dissimilar genre. They are not directly comparable to our own results that train on SWBDdata and test on WSJ because we have used a larger training set but done less to maximise performance. However, when reinterpreted in this way, his results are broadly compatible with our own.

3 The Data

We use four different corpora in this work, first three of which are drawn from PTB [1], the last from AMI corpus [3], labelled in our figures and tables as follows:

- PTB/WSJ (CONLL)
  This is the same data that was used in CONLL-2000: that is, sections 15-18 of the WSJ portion of the PTB as training data, and section 20 as test data.
- PTB/BROWN
  This is taken from the portion of the Brown corpus that has been labelled with syntactic annotation as part of the PTB. It consists of a collection of textual sources ranging in genre, but largely drawn from different types of fiction. For our test data, we chose one text from each of the annotated subsets representing different text genres.

- PTB/SWBD
  This is taken from the portion of the Switchboard corpus that has again been labelled with syntactic annotation as part of the PTB. We used sections 2 and 3 as training data and section 4 as test data. The switchboard corpus consists of telephone conversations between pairs of unfamiliar participants in which the participants were given an overall topic of conversation.
- AMI
  This is a portion of the section of the AMI meeting corpus that consists of four-party group discussions in which the participants simulate a workplace design team. Two short meetings (IS1008a and ES2008a) are used as test data, and two long meetings (IS1008b and ES2008b) as training data.

Because the first three of these corpora come from the PTB, we simply derive chunks for them using the scripts distributed for CONLL-2000. This leaves the AMI meeting corpus, for which no PTB portion is available. To address this gap, we hand-annotated training and test data sets from the AMI corpus for most of the chunk types used in CONLL-2000, omitting PRT, LST, and UCP because they are extremely rare. [2]

From these corpora, we used WSJ (CONLL), SWBD, and AMI for training and all of the corpora for testing. The training sets vary considerably in size, as can be seen from table 1. Note that two of the four corpora are from textual sources and two consist of transcribed speech. All four are arguably from different genres, although one would generally expect the text corpora to share more characteristics with each other than with the speech corpora, and the speech corpora to share more characteristics with each other than with the text corpora. This means, for instance, that whether one's target corpus is a text corpus or a speech corpus, it would be reasonable to expect better performance when training on another corpus of the same basic type than the opposite basic type.

Table 1. Data details (in words, 1K = 1,000)

data	training	test	size
WSJ (CONLL)	220K	49K	medium
SWBD	1,054K	223K	large
BROWN	–	20K	
AMI	13.6K	6.6K	small

4 The Classifiers

When building a chunker, the first step is to choose a classifier. There were many possible classifiers we could have chosen, and if we were interested in maximising

[2] Our annotation manual is available from the first author upon request; annotation was performed using a version of the named entity annotation tool reconfigured for chunks, from the NITE XML Toolkit, which is available at `http://www.ltg.ed.ac.uk/NITE/` or `http://sourceforge.net/projects/nite/`

performance alone, we might wish to try them all. The three below represent a reasonable spread of choices. Each comes from a different theory and is both publicly available and well-implemented.

MaxEnt/MXPOST. [3] This is a classical statistical POS taggers based on maximum entropy theory [15]. Although it was written in Java about ten years ago, it is fast and demands less computing resource than the other two.

SVM/YAMCHA-TinySVM. [4] This classifier is based on support vector machines (SVM) [16]. An SVM-based chunker won the first place in the CONLL 2000 task [10]. This particular implementation produced even better result than that [11].

CRF/CRF++. [5] This is one of the post-CONLL models that achieve better performance CRF [17], based on Conditional Random Fields theory [18]. Though the experiments were carried out on NP chunking, the performance is comparable with other chunkers on the same task.

For the features, we refrain from using some complex ones: for words, we use unigram of current word and two words before and after current word; for chunk tags, we use unigram of last two tags.[6] We did not use POS information so as to make things less complicated because for any new data there will be neither directly available tagger, nor manual annotation. The reason we choose such simple features is, on the one hand, to ensure fair comparison across classifiers,[7] and on the other, to make computation as easy as possible, keeping in mind limited resources available for online processing in some real-world applications.

5 Experimental Results

We trained a couple of chunkers with the above three classifiers and simple feature configuration on the training sets of three corpora. And all the chunkers were tested on the four test sets (with additional selection of BROWN as test data). The results are given in Table 2.

First we can see from the table that despite its relatively modest computational demands, the maximum entropy classifier performs best most of the time when the test and training data come from different sources. When the test and training data come from the same source, the result of the maximum entropy classifier is very close to the best results from the CRF one.

[3] Available at `ftp://ftp.cis.upenn.edu/pub/adwait/jmx/jmx.tar.gz`

[4] Available at `http://chasen.org/~taku/software/yamcha/` and `http://chasen.org/~taku/software/TinySVM/`

[5] Available at `http://chasen.org/~taku/software/CRF++/`

[6] It was not untill very late that we found that features for MXPOST target tags are unigram of last tag and bigram of last two tags. But the result from this is only slightly better than that using unigrams of last two tags. So the results and discussions below should still hold though the comparisons are not strictly fair.

[7] This is the default feature manipulation for MXPOST. It is possible to extend through concatenation, like [8].

Table 2. Chunking performance (F1) for three classifiers trained on three training corpora, tested on the four test sets

	MaxEnt	SVM	CRF
Training on WSJ/CoNLL			
WSJ	88.35	87.85	88.55
BROWN	83.29	80.37	81.57
SWBD	71.06	70.62	71.49
AMI	62.84	58.34	61.91
Training on SWBD			
WSJ	76.70	75.39	77.78
BROWN	82.43	79.55	81.20
SWBD	89.93	90.96	91.82
AMI	75.08	71.90	73.50
Training on AMI			
WSJ	61.41	51.54	57.29
BROWN	62.62	49.62	55.16
SWBD	70.09	66.88	69.28
AMI	81.72	78.96	82.02

It is also very clear that the chunkers work best when trained on the data from the same source. This is not beyond expectation. But there is some difference between them: F1 is around 88 for WSJ, 90 for SWBD, and 81 for AMI. This could be easily explained by the difference in the training data size. From Table 1, compared with the medium-sized WSJ (CONLL), SWBD is much larger and AMI is much smaller. Further experiments on the effect of training data size on chunking performance will be presented below (§ 6).

But it is not that easy to compare the performance for the chunkers on unmatched data (i.e., trained on the data from one source, tested on the data from others). Can we explain by genre difference in terms of the distinction between written texts and spoken dialogue? For the chunkers trained on written WSJ, the performance on BROWN is much closer than that on SWBD and AMI. For the chunkers trained on AMI data, the performance on SWBD is much closer than that on WSJ and BROWN. The spoken-written distinction can partly be supported by the chunk distributions in the data, as shown in figure 1.

But for the chunkers trained on SWBD, the performance on BROWN is much closer than that on WSJ and AMI, which could not be simply explained by the spoken-written distinction. We further calculated Kullback-Leibler divergence (or relative entropy) – a measure of distance between two distributions p and q – for the chunk distributions. It is defined as $D(p||q) = \sum_{x \in X} p(x) \log p(x)/q(x)$. For our case, $p, q \in$ [WSJ, SWBD, BROWN, AMI], x is the chunk type. The result is given in Table 3.

Since Kullback-Leibler divergence indicates the inefficiency of assuming that the distribution is q when the true distribution is p [19], and is asymmetric with regard to p and q (as can been seen from both the definition and table 3), if we look at the table horizontally, we can see that it would be more efficient

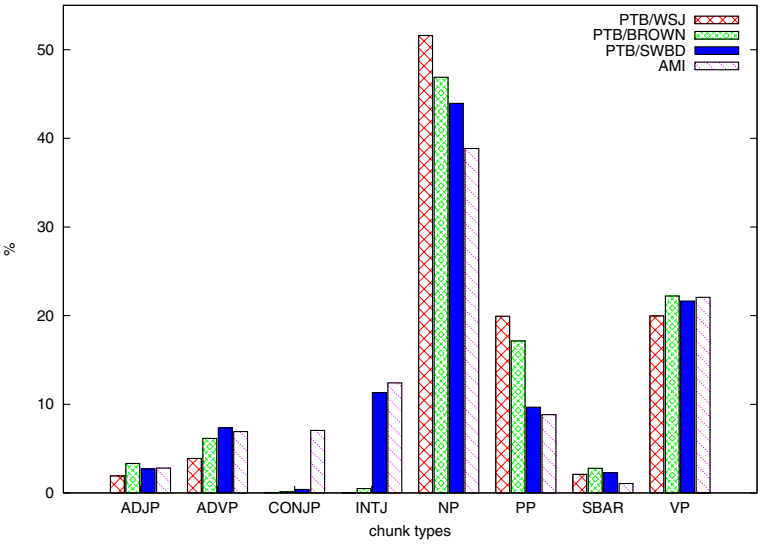


Fig. 1. Chunk distribution in the data

Table 3. Kullback-Leibler divergence for chunk distributions

q ↓ p →	WSJ (CONLL)	BROWN	SWBD	AMI
WSJ (CONLL)	0.00	0.02	0.61	0.98
BROWN	0.02	0.00	0.27	0.54
SWBD	0.18	0.12	0.00	0.15
AMI	0.27	0.21	0.06	0.00

- to approximate BROWN with WSJ than SWBD and AMI;
- to approximate WSJ with BROWN than SWBD and AMI; and
- to approximate SWBD with AMI than WSJ and BROWN.

And although it seems more efficient to approximate BROWN with SWBD than WSJ and AMI, it is similarly efficient (or inefficient) to approximate any of WSJ, BROWN and AMI with SWBD. This explains why for the chunkers trained on SWBD the performance on BROWN is much closer than that on WSJ and AMI, instead of that the performance on AMI should be closer than that on WSJ and BROWN. If we look at the table vertically, we can see that it would be more efficient

- to approximate WSJ with BROWN than with SWBD or AMI,
- to approximate BROWN with WSJ than with SWBD or AMI,
- to approximate SWBD with AMI than with WSJ or BROWN, and
- to approximate AMI with SWBD than with WSJ or BROWN.

This should help on how to choose from relavant data when we have to build chunkers without matched data.

6 Estimating Data Requirements

So far we have investigated the effects of employing different training corpora on chunking performance. There is another open and similarly important question: when annotating new data from the target corpus, how much data is required to reach a given level of performance? Having the answer to this question, together with the cost of annotation, would aid those planning work that relies on chunking technology.

To answer the question, we run an experiment for WSJ and SWBD data that starts with 10K words of training data and add 10K words at each iteration, testing performance at each data round. The experiment uses the same data set as before for SWBD, which we again label as SWBD, but instead of restricting ourselves to the 220K words of WSJ training data that were employed in CONLL, here we used the full 1,173K words available from the PTB. We chose our maximum entropy classifier for this investigation because it provided the best all-round performance for the lowest computational costs. But this time we ran two feature set versions: our original, plus one that concatenates POS to the word features used. The POS tags do not come from some automatic tagger, but from the PTB annotation. This is to avoid unnecessary uncertainty. But in many cases we will have to use the output of some POS tagger. The results are shown in Figure 2.

In the figure, the learning curves are all quite similar. POS tags help for WSJ data when the training data set is small, but these gains nearly disappear as

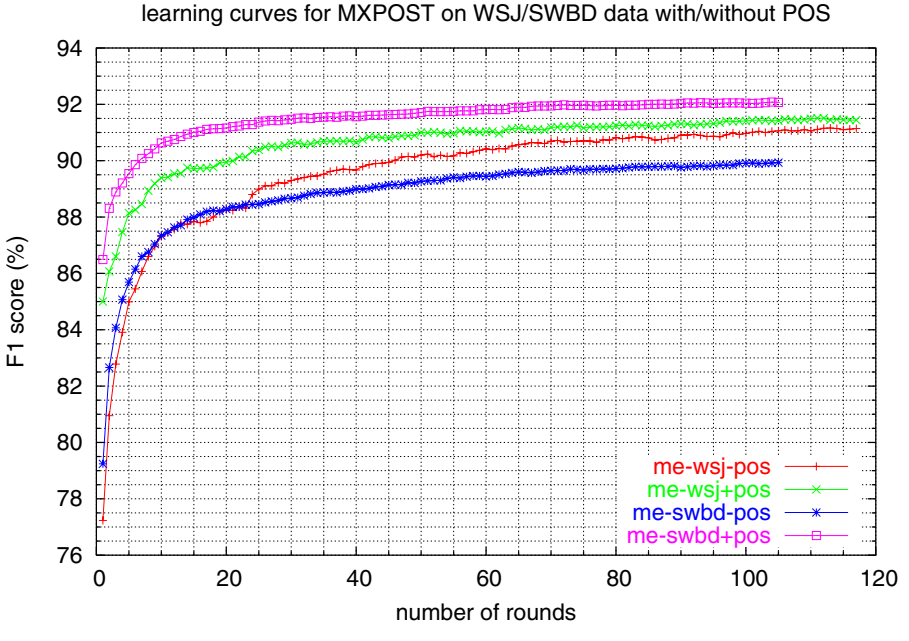


Fig. 2. MXPOST on WSJ (full) and SWBD, 10K words per round

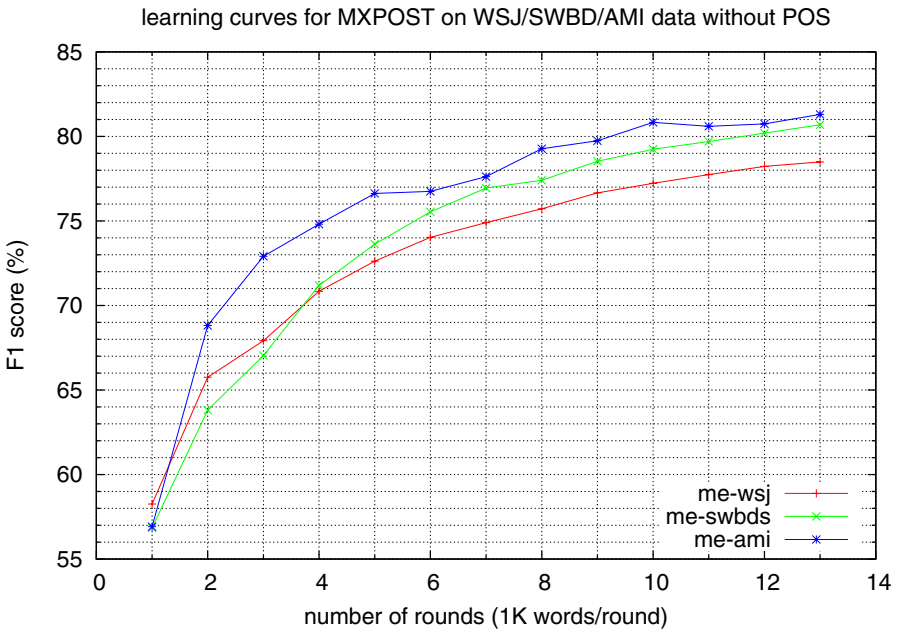


Fig. 3. MXPOST on WSJ, SWBD and AMI, 1K words per round

the training data set size increases. However, for the SWBD corpus, the gains remain almost constant throughout the curve. The figure shows that after 20-25 iterations, or 200-250K words, the potential performance gains of annotating more data are really quite low.

Although the amount of training data that we used was in this range or higher for the WSJ and SWBD corpora, our AMI training data set was much smaller than this, as it consisted of 13.6K words. Figure 3 shows the result of a similar experiment intended to blow up the initial section of the learning curve by having the iterations start at 1K words and add 1K words per round. This time we exclude the POS tagged conditions, but include a curve for the AMI data. The figure shows that the learning curve for the AMI data is very similar to those for the other corpora, at least in this initial section, and gives no reason to believe that the relationship between performance and data requirements is any different for this corpus.

7 Discussion

We have presented an empirical study on cross-corpora syntactic chunking, training on three different corpora and testing on four. We also plot the learning curves in order to estimate how much data would be enough for training a chunker. Our work is a promising beginning for the deployment of chunking in a wider range of speech and language technologies.

Our experimentation shows

- that if the very highest performance is required, there is no substitute for training chunkers on plentiful annotation from the target corpus.

  In our experiments on data from the AMI meeting corpus, the best performance we attained using training data from a different corpus could be easily beaten by using annotation from the target corpus for an extremely modest amount of data (ca 4.3K words).[8] This also echoes a similar statement from statistical parsing [20, p.200]: "a small amount of matched training data appears to be more useful than a large amount of unmatched data." But annotation is expensive. Performance gains diminish greatly once 200-250 K words of annotation are available for the target corpus, suggesting there is little point in investment beyond this.

- that if there is no annotation from the same source available, but only some annotated data from other sources (maybe in other form, like syntactic trees in the PTB), then reuse a more similar data would be more beneficial. The performance levels that can be attained by training on existing data from other corpora may well suffice for many end applications, particularly those in speech where imperfect speech recognition means that safeguards against incorrect interpretation must be built in anyway.

- that chunking spontaneous speech does not raise any particular difficulties or require any change of chunking strategies previously derived for newspaper texts. Actually it is the distributional difference that matters more to chunking than the distinction between written text and spoken dialogue where there is more noise from disfluencies. Hence chunking is very much noise-tolerant. This further confirms previous work on the effect of noisy materials on chunking [9].

There are still some open problems. In order to determine the best strategies for developing a chunker on new types of data, we must develop some measure (like K-L divergence) for the distance between source and target corpora that predicts how well a chunker trained on the source will perform on the target. In future work we should also consider the possibility of training chunkers using data drawn from multiple sources and determine how to predict the amount of corpus-specific annotation required to raise performance to whatever level an application requires.We need work on some general framework for domain/genre adaption, like [21]. Another more challenging task would be to find out how semi-supervised techniques could help chunking through learning from additional unlabelled data, like [14].

[8] From Table 2, the best performance for AMI data while training on non-AMI data is F1=75.08 (training on SWBD with MXPOST). This performance could be achieved by the chunker trained on AMI data at the amount of 4.3K words or so, as can be seen from Figure 3.

Acknowledgement

Special thanks go to Adwait Ratnaparkhi and Taku Kudo for their providing the wonderful softwares publicly available.

References

[1] Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: the penn treebank. Computational Linguistics **19**(2) (1993) 313–330
[2] Tjong Kim Sang, E.F., Buchholz, S.: Introduction to the conll-2000 shared task: Chunking. In Cardie, C., Daelemans, W., Nedellec, C., Tjong Kim Sang, E., eds.: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal (2000) 127–132
[3] Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., , Wellner, P.: The AMI meeting corpus: A pre-announcement. In: Proceedings of 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms. (2005)
[4] Carreras, X., Màrquez, L.: Introduction to the CoNLL-2005 shared task: Semantic role labeling. In: Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005), Ann Arbor, Michigan, Association for Computational Linguistics (2005) 152–164
[5] Abney, S.: Parsing by chunks. In Berwick, R.C., Abney, S.P., Tenny, C., eds.: Principle-Based Parsing: Computation and Psycholinguistics. Kluwer Academic Publishers, Boston (1991) 257–278
[6] Abney, S.: Partial parsing via finite-state cascade. Natural Language Engineering **2**(4) (1996) 337–344
[7] Ramshaw, L., Marcus, M.: Text chunking using transformation-based learning. In Yarovsky, D., Church, K., eds.: Proceedings of the Third Workshop on Very Large Corpora. (1995) 82–94
[8] Osborne, M.: Shallow parsing as part-of-speech tagging. In Cardie, C., Daelemans, W., Nedellec, C., Tjong Kim Sang, E., eds.: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal (2000) 145–147
[9] Osborne, M.: Shallow parsing using noisy and non-stationary training material. Journal of Machine Learning Research **2** (2002) 695–719
[10] Kudo, T., Matsumoto, Y.: Use of support vector learning for chunk identification. In Cardie, C., Daelemans, W., Nedellec, C., Tjong Kim Sang, E., eds.: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal (2000) 142–144
[11] Kudo, T., Matsumoto, Y.: Chunking with support vector machines. In: Proceedings of NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001, Morristown, NJ, USA, Association for Computational Linguistics (2001) 1–8
[12] Zhang, T., Damerau, F., Johnson, D.: Text chunking based on a generalization of winnow. Journal of Machine Learning Research **2** (2002) 615–637
[13] Carreras, X., Màrquez, L., Castro, J.: Filtering-ranking perceptron learning for partial parsing. Machine Learning **60** (2005) 41–71
[14] Ando, R., Zhang, T.: A high-performance semi-supervised learning method for text chunking. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, Michigan, Association for Computational Linguistics (2005) 1–9

[15] Ratnaparkhi, A.: A maximum entropy part-of-speech tagger. In Brill, E., Church, K., eds.: Proceedings of the Conference on Empirical Methods in Natural Language Processing 1996. (1996) 133–142
[16] Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons (1998)
[17] Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Morristown, NJ, USA, Association for Computational Linguistics (2003) 134–141
[18] Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 282–289
[19] Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley-Interscience, New York, NY, USA (1991)
[20] Gildea, D.: Corpus variation and parser performance. In Lee, L., Harman, D., eds.: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing. (2001) 167–202
[21] Daumé III, H., Marcu, D.: Domain adaptation for statistical classifiers. Journal of Artificial Intelligence Research (2006) conditionally accepted.

Multistream Recognition of Dialogue Acts in Meetings

Alfred Dielmann and Steve Renals

Centre for Speech Technology Research
University of Edinburgh,
Edinburgh EH8 9LW, UK
{a.dielmann,s.renals}@ed.ac.uk

Abstract. We propose a joint segmentation and classification approach for the dialogue act recognition task on natural multi-party meetings (ICSI Meeting Corpus). Five broad DA categories are automatically recognised using a generative Dynamic Bayesian Network based infrastructure. Prosodic features and a switching graphical model are used to estimate DA boundaries, in conjunction with a factored language model which is used to relate words and DA categories. This easily generalizable and extensible system promotes a rational approach to the joint DA segmentation and recognition task, and is capable of good recognition performance.

1 Introduction

This paper is concerned with automatically extracting some facets of the discourse structure of multiparty meetings. In particular we are concerned with the automatic recognition of *dialogue acts* (DAs). Each utterance in a transcription of a meeting can be associated to a dialogue act (or several dialogue acts) describing the function that the utterance serves in the conversation. This generic definition leaves space for several different DA coding schemes, that may be targeted on different aspects of the conversational process or simply characterised by a different number of sub-categories.

In this work we are interested in a DA dictionary composed of a few generic DA categories [1]. Classes of dialogue act in this scheme, which was obtained from the richer Meeting Recorder Dialogue Act (MRDA) annotation scheme [2], consisted of *statements*, *questions*, *fillers*, *back-channel* and *disruptions*. Those broad DA categories can be seen as the basic building blocks of a conversation, and thus they may be employed in modelling more complex meeting behaviours, such as meeting phases, or to enhance processes such as language modelling for automatic speech recognition or topic detection.

The DA recognition process is composed of two main steps: segmentation and tagging. The first step consists of subdividing the sequence of transcribed words in terms of DA segments. The goal is to segment the text into utterances that have approximately similar temporal boundaries to the annotated DA units. The second step of DA tagging takes DA segments as input and classifies them as one of the five DA classes listed above. These two steps may be performed either sequentially (segmentation followed by classification) or jointly (both tasks carried out simultaneously by an integrated system). In this paper we focus on the joint segmentation and classification approach, using

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 178–189, 2006.

trainable statistical models: dynamic Bayesian networks (DBNs). We note that the full DA recogniser can be forced to operate on pre-segmented data, hence acting as a simpler DA tagger. Alternatively, by discarding the DA tags the system may be employed for the segmentation task alone.

The paper is organised as follows. The next section reviews some DA recognition works carried out on natural multi-party meetings, with a particular focus on the ICSI meeting corpus, described in section 3. Section 4 outlines our DA recognition framework and its components: the feature extraction process (section 5), the DA factored language model (section 6), and the generative DBN-based infrastructure (section 7). Experiments using this framework and five different setups are reported in section 8. Finally, section 9 proposes a brief summary and concludes with some final notes.

2 Related Work

Stolcke et al. [3] provide a good introduction to dialogue act modelling in conversational telephone speech, a domain with some similarities to multiparty meetings. Dialogue acts may be modelled using a generative hidden Markov model [4], in which observable feature streams are generated by hidden state DA sequences. Most DA recognizers are based on statistical language models evaluated from transcribed words, or on prosodic features extracted directly from audio recordings. Various language models have been tried, including factored language models [5], although any kind of trainable language model can be adopted. Prosodic features provide a large range of opportunities, with entities such as duration, pitch, energy, rate of speech and pauses being measured using different approaches and techniques [6,7]. Other features, such as speaker sex, have also been usefully integrated into the processing framework.

The most likely sequence of dialogue acts is inferred from the lexical and prosodic data, and from a discourse model. The discourse or dialogue act grammar could be estimated using a simple n-gram model based on DA labels or more exotic language models evaluated from the distribution of DA-tags. Note that precise utterance and dialogue act boundaries are often assumed to be known a priori as part of the DA annotation (tagging task). When this information is not available (recognition task), it is estimated by employing automatic segmentation algorithms.

Ang et al. [1] addressed the automatic dialog act recognition problem using a sequential approach, in which DA segmentation was followed by classification of the candidate segments. Promising results were achieved by integrating a boundary detector based on *vocal pauses* with a hidden-event language model HE-LM (a language model including dialogue act boundaries as pseudo-words). The dialogue act classification task was carried out using a maximum entropy classifier, together with a relevant set of textual and prosodic features. This system segmented and and tagged DAs in the ICSI Meeting Corpus, with relatively good levels of accuracy. However results comparing manual with automatic ASR transcriptions indicated that the ASR error rate resulted in a substantial reduction in accuracy.

Using the same experimental setup, Zimmermann et al. [8] proposed an integrated framework to perform joint DA segmentation and classification. Two lexical based approaches were investigated, based on an extended HE-LM (able to predict not only the

DA boundaries but also the DA type), and a HMM part of speech inspired approach. Both these approaches provided slightly lower accuracy when compared with the two-step framework [1], but this may be accounted by the lack of prosodic features.

Ji et al. [9] propose a switching-DBN based implementation of the HMM approach outlined above, which they applied to dialogue act tagging on ICSI meeting data. They also investigated a conditional model, in which the words of the current sentence generate the current dialog act (instead of having dialogue acts which generate sequence of words). Since this work used only lexical features, and a large number of DA categories (62), a direct comparison with the results provided by [1] is not possible.

Venkataraman et al. in [10] proposed an approach to bootstrap a HMM-based dialogue act tagger from a small amount of labeled data followed by an iterative retraining on unlabeled data. This procedure enables a tagger to be trained on an annotated corpus, then adapted using similar, but unlabeled, data. The proposed tagger makes use of the standard HMM framework, together with dialogue act specific language models (3-grams) and a decision tree based prosodic model. The authors also advance the idea of a completely unsupervised DA tagger in which DA classes are directly inferred from data.

3 Annotated Data

The experiments reported in this paper use the ICSI Meeting Corpus [11]. This corpus consists of 75 multiparty meetings recorded with multiple microphones: one head-mounted microphone per participant and four tabletop microphones. Each meeting lasts about one hour and involves an average of six participants, resulting in about 72 hours of multichannel audio data. The corpus contains human-to-human interactions recorded from naturally occurring meetings. Moreover, having different meeting topics and meeting types, the data set is heterogeneous both in terms of content and structure.

Orthographic transcriptions are available for the entire corpus, and each meeting has been manually segmented and annotated in terms of Dialogue Acts, using the ICSI MRDA scheme [2]. The MRDA scheme is based on a hierarchy of DA types and subtypes (11 generic tags and 39 specific sub-tags), and allows multiple sub-categorizations for a single DA unit. This extremely rich annotation scheme results in more than a thousand unique DAs, although many are observed infrequently. To reduce the number of sparsely observed categories, we have adopted a reduced set of five broad DA categories [1,8]. Unique DAs were manually grouped into five generic categories: statements, questions, backchannels, fillers and disruptions. The distribution of these categories across the corpus is shown in table 1. Note that statements are the most frequently occurring unit, and also the longest, having an average length of 2.3 seconds (9 words). All the other categories (except backchannels which usually last only a tenth of a second) share an average length of 1.6 seconds (6 words). An average meeting contains about 1500 DA units.

The corpus has been subdivided into three data sets: training set (51 meetings), development set (11 meetings) and test set (11 meetings). All our experiments were conducted on the same dataset subdivision proposed by Ang et al.[1] in order to have directly comparable results.

Table 1. Distribution of DA categories by % of the total number of DA units and by % of corpus length

Category	% of total DA units	% of corpus length
Statement	58.2	74.5
Disruption	12.9	10.1
Backchannel	12.3	0.9
Filler	10.3	8.7
Question	6.2	5.8

4 Methodology

Our framework for the integrated DA recogniser uses a generative approach composed of four main blocks: a Factored Language Model (FLM, section 6), a feature extraction component (section 5), a trigram discourse model, and a Dynamic Bayesian Network (section 7). The FLM is used to map sequences of words into DA units, and is the main component of the tagger. The discourse model consists of a standard trigram language model over DA label sequences[1]. Note that our DA tagger uses only lexical information and a discourse model. Experiments using both the reference orthographic transcription and the output of automatic speech recognition (ASR) have been carried out. The automatic transcription was provided by the AMIASR team and generated through an ASR system similar to the one outlined in [12] (word error rate of about 29%). A set of six continuous features are used for DA segmentation purposes, together with part of a DBN model. This graphical model also plays a crucial role in the tagging process and acts as the master control unit for the entire recognition process.

5 Features

A vector of six continuous word related features was extracted from audio recordings and orthographic transcriptions.

Mean and variance of F0. Fundamental frequency (F0) was estimated using the ESPS pitch tracking algorithm get_f0[2] and sampled every 10 msec. The word temporal boundaries provided by the transcription [3] were then used to estimate the mean and variance of F0 for each word. Mean F0 was subsequently normalised against the speaker average pitch in order to have a participant independent feature.

RMS energy. Average root mean square energy was estimated for each word W_i and then normalised by both the average channel energy (in order to compensate for factors such as channel gain and microphone position) and the mean energy for all tokens of word W_i.

[1] Estimated using the SRILM toolkit, available from http://www.speech.sri.com/projects/srilm/

[2] Available from http://www.speech.kth.se/snack/

[3] Note that word boundaries are estimated automatically through forced alignment between acoustic models and orthographic transcriptions, thus are characterised by a relevant amount of uncertainty.

Word length. This is the word duration normalised by the mean duration for that word computed on the entire dataset. Therefore the resulting entity is inversely proportional to the rate of speech, neglecting estimation errors.

Word relevance. The word relevance was computed to be the ratio between local term frequency within the current document and absolute term frequency across the whole meetings collection. Terms which are more relevant for the current meeting will assume scores well above the unity.

Pause duration. Interword pauses were estimated using word boundary times obtained from aligning the transcription with the acoustic signal, and re-scaled in order to have a unitary range. Note that long pauses between words may highlight sentence boundaries and thus be a strong cue to DA segmentation. In fact pause related features have already been successfully employed in several DA segmentation frameworks (section 2).

6 Factored Language Models

Factored Language Models (FLMs) [13] are a generalisation of class-based language models in which words and word-related features are bundled together. The factors in an FLM may include word-related features such as part of speech, relative position in the sentence, stem, and morphological class. Indeed, there is no limit to the number of possible factors. In the FLM perspective even the words themselves, are usually considered one of the factors. Class based language models may be interpreted as a 2-factor FLM, in which words are bundled with classes.

Given a word f_t^0 and $k-1$ features $f_t^1, f_t^2, \ldots, f_t^{k-1}$, a sentence can be seen as sequence of these factor vectors $v_t \equiv \{f_t^0, f_t^1, \ldots, f_t^k\}$. As for standard language models, the goal of FLMs is to factorise the joint distribution $p(v_1, v_2, \ldots, v_n)$ as a chain product of conditional probabilities in the form $p(v_t \mid v_{t-1}, ..., v_{t-n})$. Since words have been replaced by vectors of factors, each conditional probability is now a function of these factors: $p(f_t^0, f_t^1, \ldots, f_t^k \mid f_{t-1}^0, f_{t-1}^1, \ldots, f_{t-1}^k, f_{t-2}^0, \ldots, f_{t-2}^k, \ldots, f_{t-n}^{0:k})$.

In order to build a good FLM it is necessary to choose the optimal factorisation (analogous to the structure learning problem in graphical models) and a backoff strategy to cope with data sparsity. Note that backoff is usually operated by dropping one or more factors from a Conditional Probability Table (CPT) in favour of a simpler conditional distribution (and smaller CPT), reiterating this procedure several times. Often multiple backoff paths (strategies) are feasible and it is even possible to concurrently follow all of them by adopting a generalized parallel backoff [5].

In order to model the relationship between words and DAs we have adopted a FLM based on three factors: words, DAs and the position of each word in the DA unit. Each word w_t is part of a DA unit and is characterised by the DA label d_t. Moreover each DA segment has been subdivided in blocks of five words: if w_t is one of the first five words the position factor n_t will be equal to one, if w_t belongs to the second block $n_t = 2$, and so on. The adopted language model is defined by a product of conditional probabilities $p(w_t \mid w_{t-1}, n_t, d_t)$. Note that considering only the word factor w_t the proposed FLM could be compared to a bigram since only the relation between w_t and w_{t-1} is taken into account. When backoff is required the first term to be dropped is the previous word

w_{t-1}, leading to the backoff model $p(w_t \mid n_t, d_t)$. If a further backoff is required, the DA tag d_t will be dropped resulting in the simpler model: $p(w_t \mid n_t)$. We use Kneser-Ney discounting to smooth both the backoff steps.

In order to compare different FLM candidates, instead of comparing their perplexities, we have defined a simplified *DA tagging* task. We compare FLMs by measuring their ability to assign the correct DA label to unseen DA units. This preliminary evaluation was conduced by enhancing the FLM section of the SRILM toolkit [14] with a simple decoder, able to label each DA unit (sentence) with the most likely DA tag (factor label from a list of possible options).

The above described FLM, after training on the 51 meeting training set, was able to perform DA labeling on the 11 development set meetings with an accuracy of 69.7% using reference transcriptions and 63.4% using automatic transcriprions (70.9% and 63.6% on the 11 meetings from the test set). Replacing for example the word position factor n_t with part-of-speech tags p_t (automatically labeled by using a POS tagger trained on Broadcast News data) the accuracy on manual transcriptions fell to 61.7% (63.5% on the test set). Building the model $p(w_t \mid w_{t-1}, m_t, d_t)$, where m_t represents the information about the meeting type, the recognition rate rose to 68.2% (68.8% on the test set). A model including each of n_t, p_t and m_t with three backoff steps had slightly lower recognition rates of 67.7% on the development set and 68.2% on the test set.

7 Generative DBN Model

Bayesian Networks (BNs) are examples of directed acyclic Graphical Models (GMs). GMs represent a unifying concept in which probability theory is encapsulated inside the formalism of graph theory. Random variables are associated to nodes, and statistical independence between two random variables is represented by the lack of a connecting arc between the corresponding nodes. To model time series or data sequences, the BN formalism has been generalised into the Dynamic Bayesian Network (DBN) concept. A DBN is a collection of BNs where a single BN, with private intra-frame relations among variables, is instantiated for each temporal frame, and a set of inter-frame arcs is defined. Those connections between nodes of adjacent BNs explicitly describe the flow of time and help highlighting the temporal structure of each time-series.

A DBN is a modular and intuitive representation which provides a common underlying formalism [15] for models including Kalman filters, Hidden Markov Models, coupled HMMs and hierarchical HMMs among the others. Note that since the DBN formalism is dual to a well defined mathematical theory, a unique set of tools and techniques can be developed to perform inference, model learning and decoding of any DBN model. The Graphical Model ToolKit (GMTK) [16], for example, provides a formal language to describe DBNs and a common set of tools to experiment with them. Thus this toolkit has been adopted as the main development package for all the DBN related experiments described in this work. As anticipated in section 4 the DA recognition process is coordinated by a generative DBN based model. The overall model is depicted in figure 1. The node Y_t represents the continuous observable feature vector outlined in section 5 (associated to the word W_t). E is a binary variable that switches from zero to one when a DA boundary is detected. Since the node W_t represents a word,

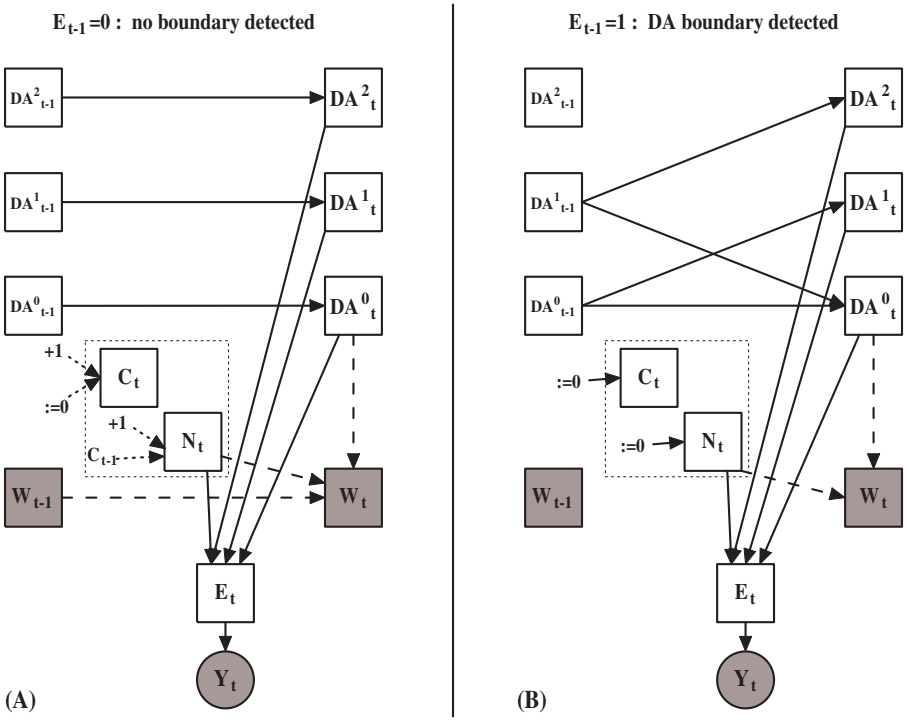


Fig. 1. Overview of the DBN model for the integrated Dialogue Act recogniser. The model's topology depends on the state of the boundary detector E_{t-1} during the previous frame: the model's graph within a DA segment has been depicted on the left(A). The right side of the picture (B) shows the new topology immediately after a DA boundary detection. Shaded square nodes represent observable discrete variables, unshaded squares correspond to hidden discrete variables, and shaded circles are associated with continuous observations. Dotted arcs are not really part of the DBN: they symbolise relationships implied by the FLM.

a DA unit can be interpreted as a sequence of words $W_{t-k},\ldots,W_{t-2},W_{t-1},W_t$ with a DA label DA^0 ($DA^0_{t-j} = DA^0$, $\forall j \in [0,k]$). DA^1 will contain the label of the previous DA unit, and DA^2 will go one more step back on the DA recognition history. C is a cyclical counter (from 0 to 5 and back to 0,1,2,...) which is used to count blocks of five words, and N accumulates the encountered word-blocks. Note that since the model's topology changes according to the state of the switching variable E_{t-1}, this is an example of a Bayesian multi-net [17].

Figure 1(A) shows the model's topology when a DA boundary has not been detected (intra-segment phase: $E_{t-1} = 0$). The current DA label DA^0 is responsible for the current sentence $W_t, W_{t-1},\ldots,W_{t-k}$ and the joint sentence probability is estimated through the FLM $p(W_t \mid W_{t-1}, N_t, DA^0_t)$ introduced in section 6. Note that FLMs are fully supported by GMTK, which will automatically take care of the backoff procedure whenever required. The word block counter N needed by the FLM is automatically incremented whenever the cyclical word counter C reach the fifth word (word block dimension

defined in section 6). All the DA label related nodes DA_t^k are simply copied from the previous temporal slice ($DA_t^k = DA_{t-1}^k$ with $k = 0, 1, 2$) since a new DA segment has not yet been recognised.

The state of the end boundary detector E is directly related to the word block counter N and the DA label history DA_t^k through a discrete CPT which is learned during training. The two states of E are linked to continuous feature vectors Y by two sets of Gaussian Mixture Models. Nodes E and Y (together with the associated CPT and GMMs) are fully responsible for the DA segmentation process. If the DA boundaries are known a priori, they can be injected into the model by making E an observable node, and the resulting system will operate as a DA tagger.

If during the previous frame $t-1$ a DA boundary has been detected, the model will be switched to the topology shown in figure 1(B) (inter-segment phase: $E_{t-1} = 1$). Since a new DA unit has been detected at the end of the previous frame $t-1$, both the counters C and N will be set to zero, and the FLM is forced to restart with a new set of estimations. The DA recognition history is updated by copying DA_{t-1}^1 into DA_t^2 and DA_{t-1}^0 into DA_t^1. The new DA hypotheses will be generated by taking in account the current DA label DA_{t-1}^0 and the previous one DA_{t-1}^1 through a trigram language model $p(DA_t^0 \mid DA_{t-1}^0, DA_{t-1}^1)$ (section 4).

The graphs in figure 1 show only the BN slices that are actually duplicated for $t > 1$. During $t = 0$ all the hidden states are properly initialised and the FLM is forced to backoff to $p(W_0 \mid N_0, DA_0^0)$ since W_0 is the first word. During the second frame $t = 1$, DA_1^2 is set to zero and the discourse language model is eventually forced to backoff to a bigram.

8 Experimental Setup and Performance Measures

All the experiments have been performed on the ICSI corpus using the five DA categories and the data sets described in section 3. The system outlined in the previous sections is primarily targeted on the DA recognition task intended as joint segmentation and classification, but as explained in section 7, it is possible to provide the ground truth segmentation and evaluate the DA tagger alone.

The percentage of correctly labeled units is about 76% on reference transcriptions and about 66% on ASR output. The classification procedure is exclusively based on the lexical information (through the FLM) and on the DA language model; prosodic related features are used only for segmentation purposes. Comparing these results with those shown in section 6, we can deduce that the introduction of a trigram discourse model has resulted in an absolute improvement included between 2% (on automatic transcriptions) and 5% (on manual transcriptions).

If performance evaluation is straightforward for the DA tagging task, the same cannot be said about DA segmentation or recognition tasks. Several evaluation metrics have been proposed, but the debate on this topic is still open. In our experiments we have adopted all the performances metrics proposed by Ang et al. [1] and subsequently extended by Zimmermann et al. [8], together with a new recognition metric inherited from the speech research community. A detailed description of these metrics (NIST "Sentence like Unit" (SU) derived metrics, strict, lenient and boundary based metrics) can

Table 2. Segmentation and recognition error rates (%) of five different system configurations

		Metric	LEXICAL	PROSODY	PAUSE	ALL (REF)	ALL (ASR)
T	S	NIST-SU	93.7	83.4	48.0	**35.6**	**43.6**
E	E	DSER	83.6	90.7	51.2	48.9	58.2
S	G	STRICT	87.4	85.8	66.4	56.5	63.5
T	M	BOUNDARY	14.5	12.9	7.4	5.5	7.3
	R	SCLITE	52.7	60.7	48.8	44.6	53.5
S	E	NIST-SU	104.1	93.8	68.5	56.8	69.6
E	C	DER	86.7	92.1	62.9	61.4	72.1
T	O	STRICT	89.1	87.6	72.5	64.7	**72.5**
	G.	LENIENT	20.7	22.0	19.5	**19.7**	**22.0**

be found in [1]. The DA Error Rate (DER) and DA Segmentation Error Rate (DSER) are discussed in [8].

The speech recognition inspired metric derives from Word Error Rate but having words replaced by DA units. Recognised DA segments are firstly time-aligned against the ground truth annotation, and then the sum of substitution, deletion and insertions errors is scored against the number of reference DA units. This error metric is estimated using the publicly available tool SCLITE (part of the NIST Speech Recognition Scoring Toolkit [4]) which also provides detailed statistics on erroneous segments and significance tests. The SCLITE metric, compared with all the other recognition metrics (except the lenient one), is more focused on a correct DA classification rather than on an extremely accurate segmentation.

Table 2 shows the segmentation and recognition results on five different setups. Results are reported using all the evaluation metrics cited above. Note that all the nine adopted metrics are "error rates", thus lower numbers correspond to better performances. The proposed setups differ only in the information used to detect DA boundaries: the *Lexical* setup makes no use of continuous features (node *Y* has been removed from the DBN), the *Prosody* setup uses only five out of six features (excluding pauses), the *Pause* setup uses the pause information but not the other continuous features, the *All (REF)* and *All (ASR)* configurations exploit the full feature set. *All (REF)* reports the results achieved by training and evaluating the DA recogniser on manually annotated orthographic transcriptions, whenever in *All (ASR)* the system has been developed and tested on automatic transcriptions. Therefore in the later experiment the combination of ASR and DA recogniser constitutes a fully automatic approach, since manual annotations are not needed. Note that the *Lexical* setup makes use of the lexical information just for DA classification purposes. Boundary detection is estimated from the current DA label, the DA history and the word block counter. Therefore this setup and the lexically based systems investigated in [8] cannot be directly compared.

The adoption of prosodic and word related features made in the *Prosody* setup presents a conflicting behaviour: NIST-SU, strict and boundary metrics show an improvement over the baseline setup; while DSER, DER, lenient and SCLITE based metrics move toward higher error rates. The *Pause* setup shows a clear improvement over

[4] SCTK available from http://www.nist.gov/speech/tools/

the baseline approach under all the evaluation metrics, and proves its strength over the *Prosody* setup highlighting the importance of pause related information on the segmentation task.

The fully integrated approach (*All-REF*) is the most accurate model. The error rates are similar to the NIST-SU segmentation error rate (34.4%) and the lenient recognition error rate (19.6%) of the two step recogniser presented by Ang et al. [1] (section 2). This result suggests that, even if the two competing systems have similar segmentation performances, and the maximum entropy based DA classifier (about 80% correct classification [1]) seems to be more powerful than our generative approach, the joint segmenter+classifier framework is potentially able to outperform a sequential framework. This is even more evident with the fully automatic ASR based system (*All-ASR*) which provides a relevant improvement if compared to the sequential approach outlined in [1] (lenient recognition error rate of 25.1%). In the sequential approach the DA classifier will be able to process only one segmentation hypothesis, whereas in the joint approach multiple segmentation hypotheses are taken in account by the DA tagger. The final choice between multiple candidates will be carried out by taking the most likely sequence of DA units, intended as the optimal combination of DA boundaries and DA labels.

9 Summary and Discussion

We have investigated the dialogue act recognition task in multiparty conversational speech, by applying a joint segmentation and tagging approach on natural meetings (ICSI meeting recordings). The proposed system makes use of a heterogeneous set of technologies: a graphical model, a factored language model and some continuous features. The graphical model, implemented as a DBN-based multi-net, oversees the whole recognition process. The proposed model adopts a generative paradigm for the DA tagging task and performs DA segmentation through a feature based architecture. DA tagging is performed using a factored language model over DA labels and word positions, together with a discourse language model. DA segmentation is obtained by exploiting both the DA discourse model and a set of six continuous features extracted from audio recordings and orthographic transcriptions.

The joint DA recognition approach, if compared to a sequential one, provides a clearer view of the addressed problem and an intuitive strategy to its solution. The integrated approach encourages the reuse of common resources such as features and model parts. For example our graphical model shares the DA discourse model between the two subtasks (segmentation and classification), and makes the word block counter required by the FLM available for segmentation purposes (duration model). Furthermore the joint approach operates on a wider search space (producing joint sequences of segmentation boundaries and DA labels based on a trigram discourse model), and thus it is potentially capable of better recognition results. For example the results achieved in our reference transcription based experiments are similar to the sequential DA recognition approach proposed by Ang et al. [1], even though the maximum entropy DA classification approach chosen by the former work provides a 5% higher tagging accuracy. The advantage of a joint approach is substantial when manual orthographic transcriptions are replaced by imperfect automatic transcriptions. The lenient DA recognition

error rate is degraded by only 2.3% and the comparison between sequential and joint approach is in favour of the latter one.

In the near future it is our intention to evaluate the present system on the new AMI meeting corpus [18] and on a richer DA annotation scheme. Moreover we would like to improve both DA classification and DA segmentation by improving the factored language model and by adopting a wider set of multimodal features.

Acknowledgment

We thank Matthias Zimmermann and Elizabeth Shriberg for advice on broad DA categories. This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication AMI-162).

References

1. J. Ang, Y. Liu, and E. Shriberg. Automatic dialog act segmentation and classification in multiparty meetings. *Proc. of the IEEE ICASSP*, March 2005.
2. E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. The ICSI meeting recorder dialog act (MRDA) corpus. *Proc. HLT-NAACL SIGDIAL Workshop*, April–May 2004.
3. A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, (26):339–373, 2000.
4. M. Nagata and T. Morimoto. An experimental statistical dialogue model to predict the speech act type of the next utterance. *Proc. of the International Symposium on Spoken Dialogue*, pages 83–86, November 1993.
5. J. Bilmes and K. Kirchhoff. Factored language models and generalized parallel backoff. *Proceedings of HLT/NAACL 2003*, May 2003.
6. E. Shriberg, R. Bates, P. Taylor, A. Stolcke, D. Jurafsky, K. Ries, N. Coccaro, R. Martin, M. Meteer, and C. Van Ess-Dykema. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, (41):439–487, 1998.
7. H. Hastie, M. Poesio, and S. Isard. Automatically predicting dialogue structure using prosodic features. *Speech Communication*, (36):63–79, 2002.
8. M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke. Toward joint segmentation and classification of dialog acts in multiparty meetings. *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–05)*, 2006.
9. G. Ji and J. Bilmes. Dialog act tagging using graphical models. *Proc. of the IEEE ICASSP*, March 2005.
10. A. Venkataraman, L. Ferrer, A. Stolcke, and E. Shriberg. Training a prosody-based dialog act tagger from unlabeled data. *Proc. of the IEEE ICASSP*, April 2003.
11. A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C.Wooters. The ICSI meeting corpus. *Proc. IEEE ICASSP*, April 2003.
12. T. Hain, M. Karafit, G. Garau, D. Moore, V. Wan, R. Ordelman, and S. Renals. Transcription of conference room meetings: an investigation. *Proc. Interspeech 2005 - Eurospeech, Lisbon*, September 2005.
13. K. Kirchhoff, J. Bilmes, J. Henderson, R. Schwartz, M. Noamany, P. Schone, G. Ji, S. Das, M. Egan, F. He, D. Vergyri, D. Liu, and N. Duta. Novel approaches to arabic speech recognition - final report from the jhu summer workshop 2002. *Tech. Rep., John-Hopkins University*, 2002.

14. A. Stolcke. SRILM an extensible language modeling toolkit. *Proc. Int. Conf. on Spoken Language Processing*, September 2002.
15. K. P. Murphy. Dynamic Bayesian networks: Representation, inference and learning. *Ph.D. Thesis, UC Berkeley, Computer Science Division*, July 2002.
16. J. Bilmes and G. Zweig. The Graphical Model ToolKit: an open source software system for speech and time-series processing. *Proc. IEEE ICASSP*, Jun. 2002.
17. J.A. Bilmes. Dynamic bayesian multinets. *Proc. Int. Conf. on Uncertainty in Artificial Intelligence*, 2000.
18. J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI meeting corpus: A pre-announcement. *Proc. Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI–05)*, 2006.

Text Based Dialog Act Classification for Multiparty Meetings

Matthias Zimmermann[1], Dilek Hakkani-Tür[1],
Elizabeth Shriberg[1,2], and Andreas Stolcke[1,2]

[1] International Computer Science Institute (ICSI), Berkeley CA 94704, USA
[2] SRI International, Menlo Park CA 94025, USA

Abstract. This paper evaluates the performance of various machine learning approaches and their combination for text based dialog act (DA) classification of meetings data. For this task, boosting and three other text based approaches previously described in the literature are used. To further improve the classification performance, three different combination schemes take into account the results of the individual classifiers. All classification methods are evaluated on the ICSI Meeting Corpus based on both reference transcripts and the output of a speech-to-text (STT) system. The results indicate that both the proposed boosting based approach and a method relying on maximum entropy substantially outperform the use of mini language models and a simple scheme relying on cue phrases. The best performance was achieved by combining individual methods with a multilayer perceptron.

1 Introduction

Dialog acts (DAs) represent the functional building blocks of conversations [1] and the classification of dialog acts corresponds to assigning DA types to the individual utterances. How these DA types are defined depends on the experimental setup or the goal of the application. A variety of DA tagging schemes have been proposed in the literature. While earlier schemes were designed for specific applications in mind more recent approaches try to cover unconstrained conversations. Examples for this range of annotation schemes are MAPTASK [2], VERBMOBIL [3], DAMSL [4], and [5]. In our study we refer to the following five classes as DA types: Statements, questions, backchannels, floorgrabbers, and disruptions. These DA types have been used in previous work [6,7] and are defined to be mutually exclusive. Backchannels are mostly single word utterances such as *uhhuh* or *yeah* that are used to indicate that the speaker should go on talking. Floorgrabbers subsume all utterances that are linked to the floor management in multiparty conversations. Frequently, floorgrabbers such as *well* or *i think* are used during an ongoing conversation to indicate that a participant would like to start talking. A speaker can also use floorgrabbers (e.g. *so*) while talking to indicate that she/he is not finished talking yet. Finally, disruptions categorize all utterances that can not be completed as intended by the speaker including self-interruptions. With these five DA types we try to achieve a balance between

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 190–199, 2006.

the desire to model dialog accurately and the resulting amount of training data per DA class. Our set of DAs might be used to support applications such as meeting browsing and summarization (where backchannels, floorgrabbers, and disruptions would be filtered out) or could help to train meeting agents that would try to interact with the other participants as naturally as possible (i.e. responds to questions, or uses floor grabbers correctly, etc.).

The goal of this paper is to compare the performance of boosting based DA classification to various text based techniques described in the literature. For this, we use a tightly controlled experimental setup that only allows the methods to access isolated utterances (i.e. the classification is based on words within a given utterance and does not make use of other knowledge sources such as the sequence of utterances or prosody).

Previous work mainly investigated single methods or variations of a single method for the classification of dialog acts. Most prominently, methods relying on word n-gram language models have been investigated in various experimental setups [8,3,9,10]. Semantic classification trees have been proposed in [11], transformation based learning was investigated in [12], and artificial neural networks were used in [13]. More recently, dynamic Bayesian networks have been proposed as well [14]. To our knowledge boosting-based classification of dialog acts has not been investigated so far, and no direct comparison of the performance for the methods investigated in this work is available.

The remainder of this paper is organized as follows. First, the four different DA classification methods are described in the following section. Experiments and results are described in Section 3, while conclusions and possible future work are provided in Section 4.

2 Methodology

All four DA classification methods described below attempt to predict the most likely DA type d^* for a given utterance $W = (w_1, w_2, \ldots w_n)$. In the next subsections the most widely used technique, based on DA-specific language models is covered first. Section 2.2 then summarizes a method relying on cue phrases [15]. A maximum entropy based approach is described in Section 2.3, and the proposed boosting based method is introduced in Section 2.4.

2.1 Mini Language Models

Dialog act-specific mini n-gram language models (Mini LMs) proposed by [8] have been widely used in previous work [3,10]. For each DA type d, an individual word n-gram LM is trained on all utterances from an annotated corpus that are tagged with the desired DA type d. This training procedure allows the Mini LMs to capture the DA specific word usage and produce DA specific likelihoods $p(W|d)$.

To classify an unknown utterance W the estimates must then be multiplied by the prior probability $p(d)$ leading to the decision rule given below.

$$d^* = \underset{d}{argmax}\, p(W|d)\; p(d) \tag{1}$$

Although this method represents a principled approach that relies on the well known domain of n-gram language modeling it has the drawback of not being trained in a discriminative way.

2.2 Cue Phrases

The second technique investigated here relies on the concept of cue phrases that correspond to word n-grams up to a specified order. The scheme has been proposed in [15] and is particularly simple to implement. During training the list of cue phrases is constructed in the following way. Initially cue phrase candidates include all word n-grams of a given corpus for $n = 1$ up to $n = 4$. For each such cue phrase C its *predictivity* $p(d|C)$ is computed that measures to which extend the presence of this cue phrase indicates the specific DA type d. For each cue phrase candidate its maximal predictivity that corresponds to the most likely DA type for the presence of this cue phrase is determined. Two thresholds are then used to obtain the final cue phrases. The first threshold requires a cue phrase candidate to be observed at least a given amount of times in the training corpus, and the second threshold only retains cue phrases that exceed a fixed minimal predictivity.

For an unknown utterance W all known cue phrases are then extracted and the DA type corresponding to the cue phrase that is associated with the highest predictivity is used to output the result d^* In our implementation we used the system including position specific cues by explicit modeling of the start and the end of utterances; see [15] for further details. A potential drawback of this method lies in its decision rule that does not generalize well to produce a score for each available DA type.

2.3 Maximum Entropy

One of the main drawbacks of the methods described above lies in their training that does not explicitly optimize the discrimination between correct and incorrect DA types for a given utterance. To take advantage of discriminative training a DA classification technique based on maximum entropy modeling was proposed in [6]. Furthermore, the maximum entropy framework allows directly estimating posterior probabilities $p(d|F)$ for a DA type d and a binary feature vector F (see [16] for an excellent introduction into maximum entropy modeling).

The DA type d^* of an unknown utterance is determined by the DA type d that maximizes the posterior probability $p(d|F)$. In our case the feature vector F is extracted from the utterance W according to [6]. As features the first two words (after removing filler words), the last two words, the initial and the final word bigram, as well as the length of the utterance is used. In contrast to [6] we do not include the first word of the following DA, as our experimental setup only considers isolated utterances.

2.4 Boosting

The fourth method, boosting, is also discriminative and is derived from a text categorization task. Boosting aims to combine "weak" base classifiers to come

Table 1. Description of the three experimental conditions. The table lists the number of DAs, words, the chance classification error rate (Chance), and corresponding word error rates (WER).

Condition	DAs	Words	Chance	WER
Reference	113,191	740,837	45.0%	-
STT Manual	103,443	683,434	42.0%	35.4%
STT Auto	85,501	653,879	35.5%	38.2%

up with a "strong" classifier. The learning algorithm is iterative, and in each iteration, a weak classifier is learned so as to minimize the training error, and a different distribution or weighting over the training examples is used to give more emphasis to examples that are often misclassified by the preceding weak classifiers. For this approach we use the *BoosTexter* algorithm described in [17], with word n-gram features, as well as features like segment length. For comparison we have also trained a BoosTexter model with the same feature set as best performing maximum entropy approach.

3 Experiments and Results

This section first describes the experimental setup. The optimization of the individual classification methods on the development sets are described in Section 3.2 and the evaluation of the methods on the test sets is provided in Section 3.3, which also investigates various combination schemes.

3.1 Experimental Setup

For all experiments reported here the experimental setup is closely linked with the one described in [6]. Of the 75 available meetings in the ICSI MRDA corpus, two meetings of a different nature are excluded (Btr001 and Btr002). From the remaining meetings we use 51 for training, 11 for development, and 11 for evaluation. The available DA types are mapped to the following five mutually exclusive types: backchannels (B), disruptions (D), floor grabbers (F), questions (Q), and statements (S). Our setup differs in a number of aspects from [6]. In contrast to [6] we use the normalized words coming from forced alignments under reference conditions instead of the unnormalized words from the meeting transcriptions. Under reference conditions we assume access to the true words provided by the transcriptions.

For the speech-to-text (STT) conditions we also make use of a better, more recently developed recognizer [18]. Instead of a 39% word error rate (WER) the new recognizer achieves a 35.4% WER based on the close talking microphones. Furthermore, we define two separate STT conditions. The first one corresponds to the STT conditions of [6] and relies on a manual segmentation of the audio input stream (STT Manual). As a more realistic setup we also include results for automatic segmentation of the audio (STT Auto) leading to a higher WER of

Table 2. Test set frequencies of the DA types under reference conditions (Ref), STT conditions based on manual segments (STT Manual), and STT conditions using automatic segments (STT Auto)

DA Type	Ref	STT Manual	STT Auto
Statements	8,918	8,642	7,740
Questions	1,164	1,108	1,004
Backchannels	1,960	1,437	220
Floor-grabbers	1,924	1,768	1,306
Disruptions	2,237	1,937	1,734

38.2%[1]. The ground truth for both STT conditions is generated by aligning the words of the reference setup (that has been annotated for the DAs) to both of the STT conditions, with the constraint that two aligned words may not occur further apart than a fixed time (1 second). Through these alignments the STT words then inherit the DA boundaries and types from the reference conditions. See Table 1 for some statistics of the experimental conditions described in the text above.

Table 2 reports the number of DAs for each DA type for the different test sets used in our experiments. From these DA type statistics the chance error rates as reported in Table 1 is computed by the relative frequency of the statements. It is interesting to observe that, under STT Auto conditions, a large amount of single word DAs that occur in isolation are missed by the automatic segmenter. This effect is particularly dramatic for backchannels for which almost 90% of the DAs are missing compared to the reference conditions. As a result, the 5-way classification problem is almost reduced to a 4-way classification problem resulting in a significantly lower chance error rate.

3.2 System Optimization

For the Mini LM approach described in [8] the DA-specific LMs were trained using interpolated Kneser-Ney smoothing [19]. The order of the n-gram models was optimized up to $n = 4$ on the development set. Significantly different results (using a sign test) were obtained for the step from unigram LMs (error rate 36.7%) to bigram LMs (error rate 27.5%). Trigram and fourgram Mini LMs performed slightly worse but not significantly different from the bigram LMs. The choice of trigrams in [10] can be explained by the fact that the Switchboard corpus contains almost twice as many utterances as the ICSI MRDA corpus. We also compared the effect on training under STT conditions versus training under reference conditions. As training of the LMs under STT conditions slightly decreased the performance of this approach the DA-specific LMs were trained under reference conditions for all experimental setups.

For the Cue Phrase method described in [15] we measured the error rates for different maximum lengths of the cue phrases. In correspondence with [15]

[1] The difference in WER of the STT Auto conditions compared to the STT Manual conditions is mostly generated by a higher number of deletions.

the best results were achieved when cue phrases up to 4-grams were used (error rate: 27.3%). In contrast to the Mini LM approach the higher order n-grams significantly helped to improve the performance over the use of bigrams only, for which an error rate of 30.4% was measured. As in the case of the Mini LM, training of the cue phrase based method worked best under reference conditions even when the models are applied under STT conditions.

In the case of the maximum entropy based method described in [6], the effect of the number of words to include at the beginning and at the end of each dialog act was investigated as well as the influence of the removal of initial filler words. As proposed in [6], best results are obtained when initial filler words are removed, although the difference in performance is not statistically significant. The number of the initial words and final words to include as features is not very critical for the performance of this method. When only the first and the last word is kept, an error rate of 22.9% is achieved under reference conditions compared to 22.6% for keeping the first two words (plus the initial word bigram) and the final two words (and the final word bigram). Adding the first three words and last three words even leads to a small degradation of the performance (22.7%). In contrast to the previous methods, a significant gain was found when the maximum entropy based models were trained under STT conditions for use under STT conditions. The model trained under reference conditions achieved a 27.8% error rate under STT Manual conditions and a 27.3% error rate under STT Auto conditions. For training under STT Manual conditions the error rates are reduced to 27.0% (STT Manual) and 24.7% (STT Auto).

With Boosting, we have not performed any optimization and ran the classifier for 1,000 iterations for each experimental condition. Using all unigrams, bigrams, and trigrams of an utterance as features results in a classification error rate of 22.1%. When the length of the utterance is included as a feature, the classification error is reduced to 21.7%. A similar error rate, 21.9% is achieved when the BoosTexter is trained on the same features as the maximum entropy based method. These results indicate that the words at the beginning and the end of an utterance carry most of the information that can be exploited by the classification scheme. As in the case of the maximum entropy based method, training of the models under STT conditions is beneficial.

3.3 Evaluation and System Combination

After the individual optimization of each DA classification method, the best performing configuration was used for evaluation on the test sets under the three available conditions. The resulting test set error rates are reported in Table 3. In correspondence with [15] we find that the performance of the approach based on cue phrases compares well with the mini LM based approach, in spite of the difference in both corpus and DA type definitions. Furthermore, it can be observed that the two approaches based on mini LM, and cue phrases perform significantly worse than the maximum entropy based approach and the classification scheme using boosting under all investigated conditions. Under reference conditions boosting outperforms the maximum entropy based approach. For the

Table 3. Comparison of the classification error rates of the different systems under reference conditions (Ref), STT conditions based on manual segments (STT Manual), and STT conditions using automatic segments (STT Auto). The results for the combination schemes are at the bottom.

System	Ref	STT Manual	STT Auto
Mini LM [8]	26.7%	29.8%	27.3%
Cue Phrases [15]	26.6%	29.6%	28.5%
MaxEnt [6]	22.5%	26.5%	23.8%
BoosTexter	21.7%	26.9%	24.2%
Simple Voting	23.8%	27.3%	24.4%
Linear Combination	21.7%	26.3%	23.6%
MLP	21.3%	26.2%	23.3%

two STT conditions, boosting performance is slightly inferior to the maximum entropy method when all word n-grams are used as features. When we train BoosTexter on the feature set of the maximum entropy method (only keep DA initial and DA final words as features) the performance becomes the same as in the case of maximum entropy.

In a first experiment a simple voting scheme was implemented that returns the DA type most frequently predicted by the different classifiers, where in case of ties, the most frequent class is chosen. According to the results in Table 3, voting did worse than either the maximum entropy approach (MaxEnt) or the boosting based method (BoosTexter). As a more sophisticated combination method, a linear interpolation of the posterior probabilities of the DA classification methods was investigated using expectation maximization for the optimized of the interpolation weights[2]. For this combination method the probabilities of both the classifier using mini LMs and the boosting based method needed to be normalized to make sure that the results would sum up to 1 while the maximum entropy based approach directly produces posterior probabilities for all possible DA types[3]. The linear interpolation performed better than the simple voting scheme and under the STT conditions linear interpolation outperformed the individual classifiers (at a 90% level of significance). Only the multilayer Perceptron (MLP) based combination of the posterior probabilities from the Mini LM, the MaxEnt and the BoosTexter was able to significantly outperform (at the 99% level) the best individual classifiers under all conditions. For this combination method a simple feed-forward network with a single hidden layer including ten hidden neurons was trained on the development sets.

In an error analysis we combined the output of the four classifiers for each DA in the test sets to determine the oracle error rates (error rate that none of the four methods predicted the correct DA type). For the four classifiers the oracle error rate under reference conditions is 14.8%, under STT Manual conditions

[2] For each experimental condition the interpolation weights were optimized on the corresponding development set.

[3] The cue phrase based approach was not considered for this experiment as this algorithm does return a probability for only the most likely DA type.

Condition	Words	Label	Predicted
Ref.	*it's the shadow*	S	S
STT	*it's it's uh*	S	D
Ref.	*where was heidelberg ...*	Q	Q
STT	*worse heidelberg ...*	Q	S

Fig. 1. Two typical classification errors under STT conditions forced by misrecognized words. Under reference conditions (Ref.) these DAs are correctly classified. S=Statement, D=Disruption, and Q=Question.

18.3%, and 17.4% under STT Auto conditions. From these relatively low error rates a significant amount (35% under reference conditions and 27% under STT conditions) of these errors is forced by the experimental setup that does not include the context of DAs and does not consider prosody. As many frequent single-word DAs (*yeah, right, ok, uhhuh*, and *huh* can occur with different DA types. A further major source of errors under STT conditions is caused by misrecognized words that lead to readable utterances (see Fig. 1). For such cases the predicted DA types seem to be correct even for the human reader and the only chance to correctly classify such utterances is by a better speech recognition engine.

4 Conclusion and Outlook

We have investigated the performance of three text based techniques described in the literature for the classification of dialog acts in multiparty meetings. In addition, we proposed the use of a boosting-based method. From the results of the experiments we found that the boosting based method performs favorably compared to the other approaches studied in this paper. Specifically, our results indicate that both the boosting based approach and the method relying on maximum entropy significantly outperform the use of mini language models and the scheme relying on cue phrases. The best performance was achieved by a combination method that involved a multilayer perceptron. It is interesting to observe that the best performing classifiers completely (maximum entropy) or mostly (BoosTexter) rely on features derived from the first two words and the last two words of a dialog act in our classification task. This finding highlights the importance of detecting correct dialog act boundaries when we consider the joint task of segmentation and classification of dialog acts [6,7].

In future work we will investigate support vector machines for classification similar to [20,21] and the integration of syntactic information such as automatically derived POS tags and prosodic features. Alternatively, the combination of dialog act classification methods described in this paper should be put into a more realistic setup that considers joint segmentation and classification.

Acknowledgment

We would like to thank Özgür Çetin and Luke Gottlieb for their contributions in the preparation of the experimental setup. This work was partly supported

by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication), by DARPA Contract NBCHD030010 through the SRI CALO project (approved for public release, distribution unlimited), NSF Awards IIS-0121396 and IRI-9619921, and the Swiss National Science Foundation through the research network IM2.

References

1. Jurafsky, D., Martin, J.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall (2000)
2. Anderson, A., et al.: The HCRC map task corpus. Language and Speech **34**(4) (1991) 351–366
3. Reithinger, N., Klesen, M.: Dialog act classification using language models. In: Proc. ICASSP. Volume 3., Rhodes, Greece (1997) 2235–2238
4. Core, M., Allen, J.: Coding dialogues with the DAMSL annotation scheme. In: AAAI Fall Symposium on Communicative Action in Humans and Machines, Cambridge, USA (1997) 28–35
5. Shriberg, E., et al.: The ICSI meeting recorder dialog act (MRDA) corpus. In: Proc. SIGDIAL, Cambridge, USA (2004) 97–100
6. Ang, J., Liu, Y., Shriberg, E.: Automatic dialog act segmentation and classification in multiparty meetings. In: Proc. ICASSP. Volume 1., Philadelphia, USA (2005) 1061–1064
7. Zimmermann, M., Stolcke, A., Shriberg, E.: Joint segmentation and classification of dialog acts in multi-party meetings. In: Proc. 31st ICASSP. Volume 1., Toulouse, France (2006) 581–584
8. Nagata, M., Morimoto, T.: First steps toward statistical modeling of dialogue to predict the speech act type of the next utterance. Speech Communication **15** (1994) 193–203
9. Warnke, V., Kompe, R., Niemann, H., Nöth, E.: Integrated dialog act segmentation and classification using prosodic features and language models. In: Proc. 5th Europ. Conf. on Speech, Communication, and Technology. Volume 1., Rhodes, Greece (1997) 207–210
10. Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Ess-Dykema, C.V., Meteer, M.: Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational Linguistics **26**(3) (2000) 339–371
11. Mast, M., et al: Dialog act classification with the help of prosody. In: Proc. ICSLP. Volume 3., Philadelphia, USA (1996) 1732–1735
12. Samuel, K., Carberry, S., Vijay-Shanker, K.: Dialogue act tagging with transformation-based learning. In: Proc. 17th Int. Conference on Computational Linguisitics. Volume 2., Montreal, Canada (1998) 1150–1156
13. Ries, K.: HMM and neural network based speech act detection. In: Proc. ICASSP. Volume 1., Phoenix, USA (1999) 497–500
14. Ji, G., Bilmes, J.: Dialog act tagging using graphical models. In: Proc. ICASSP. Volume 1., Philadelphia, USA (2005) 33–36
15. Webb, N., Hepple, M., Wilks, Y.: Dialog act classification based on intra-utterance features. CS-05-01, Dept. of Comp. Science, University of Sheffield, UK (2005)

16. Berger, A.L., Pietra, S.A.D., Pietra, V.J.D.: A maximum entropy approach to natural language processing. Computational Linguistics **22**(1) (1996) 39–71
17. Schapire, R.E., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning **39**(2-3) (2000) 135–168
18. Stolcke, A., Anguera, X., Boakye, K., Çetin, O., Grezl, F., Janin, A., Mandal, A., Peskin, B., Wooters, C., Zheng, J.: Further progress in meeting recognition: The icsi-sri spring 2005 speech-to-text evaluation system. In Renals, S., Bengio, S., eds.: Machine Learning for Multimodal Interaction: 2nd International Workshop, MLMI 2005. LNCS 3869, Springer (2006) 463–475
19. Goodman, J.T.: A bit of progress in language modeling. MSR-TR-2001-72, Machine Learning and Applied Statistics Group, Microsoft, Redmond, USA (2001)
20. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In: Proc. Pacific Symposium on Biocomputing. (2002) 564–575
21. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. Journal of Machine Learning Research **2** (2002) 419–444

Detecting Action Items in Multi-party Meetings: Annotation and Initial Experiments

Matthew Purver, Patrick Ehlen, and John Niekrasz

Center for the Study of Language and Information
Stanford University
Stanford, CA 94305, USA
{mpurver,ehlen,niekrasz}@stanford.edu

Abstract. This paper presents the results of initial investigation and experiments into automatic *action item detection* from transcripts of multi-party human-human meetings. We start from the flat action item annotations of [1], and show that automatic classification performance is limited. We then describe a new hierarchical annotation schema based on the roles utterances play in the action item assignment process, and propose a corresponding approach to automatic detection that promises improved classification accuracy while also enabling the extraction of useful information for summarization and reporting.

1 Introduction

A great deal of everyday human-human interaction takes place in meetings, and their content can be important: information is exchanged, plans and decisions are made, new tasks assigned, deadlines changed and so on. There is consequently a great deal of interest in the automatic processing, understanding, summarization and reporting of meetings. While there may be many useful outputs that could be reported, including automatically produced transcripts or notes, user studies [2,3] suggest that amongst the most important are records of the *decisions* made and the associated *action items* assigned. This paper concentrates on the automatic detection of *action items*, specific kinds of decisions common in meetings and characterized by the concrete assignment of tasks together with certain properties such as an associated timeframe and reponsible party. Our aims are firstly to detect the regions of discourse which establish action items, so that their surface form can be used as the basis of a targeted report or summary; and secondly, to identify the important properties of the action items themselves (such as the associated tasks and deadlines), so that we can work towards more concise and informative semantically-based reporting (for example, adding task specifications to a user's calendar or to-do list). Our claim in this paper is that both of these aims are facilitated by an approach which takes into account the roles which different utterances play in the decision-making process – in short, a shallow notion of discourse structure.

We first discuss an existing set of action item annotations [1] which take a simple approach, tagging the relevant utterances as action-item-related, but not

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 200–211, 2006.

distinguishing them further. We show that while these annotations allow some useful automatic detection performance even with simple classification methods, the accuracy is limited. We then describe a richer, hierarchical approach to action item annotation, in which utterances are not only tagged as being related to a particular action item, but classified according to the role they play in the process of establishing and agreeing on that action item. Our newer approach assigns relevant utterances to one or more of a small set of decision-making acts. Initial experiments indicate that significant performance improvements can be gained by using this deeper information, detecting the individual acts independently, and then detecting action items via the presence of multiple act types. As well as improving performance, this method allows important information to be extracted to enable more detailed understanding and reporting of the detected action items.

2 Background

Action Items. In institutions where group projects and collaborative problem-solving are an important element of the institution's purpose, meetings are a common (perhaps too common [4]) occurrence where important steps are taken toward achieving both individual and institutional goals. The type of communication that takes place in these meetings can include briefings, brainstorming, problem-solving, and planning, just to name a few. For the great majority of these, the ultimate goals are to share information and make group decisions.

Action items are a common form of these group decisions that make an interesting subject for investigation for a number of reasons. First, they often embody the transfer of group responsibility to that of the individual, an extremely important component of the institutional problem-solving process. Also, a common (although by no means universal) practice in meetings is to reiterate or review them during a specific period, when individuals will summarize their commitments to others to make explicit the tasks which have been assigned to them. In addition, because they are group decisions which result in individual commitment, the committed owner has a great reponsibility to the others to be sure that the commitment is properly represented, recorded, remembered, and fulfilled.

The person committing to be responsible for the action item need not be the person who actually performs the action (they might, say, delegate the task to a subordinate), but publicly commits to seeing that the action is carried out; we call this person the *owner* of the action item. Because this action is a social action that is coordinated by more than one person, its initiation is reinforced by *agreement* and uptake among the owner and other participants that the action should and will be done. And to distinguish this action from immediate actions that occur during the meeting and from more vague future actions that are still in the planning stage, an action item will be specified as expected to be carried out within a *timeframe* that begins at some point after the meeting and extends no further than the not-too-distant future. So an action item, as a type of social action, often comprises four components: a *task description*, a *timeframe*, an *owner*, and a round of *agreement* among the owner and others.

Automatic Detection. While meeting minute-takers and managers will often summarize assigned action items, and a plethora of assistive technologies are available to facilitate that, we are unaware of any previous attempt to do this automatically for spoken discourse. There is precedent in text processing: [5] attempted to identify action items in e-mails, using classifiers trained on annotations of individual sentences within each e-mail. Sentences were annotated with one of a set of "dialogue" act classes; one class `Task` corresponded to any sentence containing items that seemed appropriate to add to an ongoing to-do list. They report good inter-annotator agreement over their general tagging exercise ($\kappa > 0.8$), although individual figures for the `Task` class are not given. They then concentrated on those sentences classed as `Task`, establishing a set of predictive features (in which word n-grams emerged as "highly predictive") and achieved reasonable per-sentence classification performance (with f-scores around 0.6).

However, for multi-party spoken dialogue (as occurs in meetings), the closest related work is probably in the area of dialogue act detection, where the dialogue act taxonomy chosen often includes decision-related classes. Many such annotation schema are ultimately based on the DAMSL annotation scheme [6] which includes tags for utterances like `Action-Directive` (a sub-class of `Influencing-Addressee-Future-Action`) and `Commit` (a sub-class of `Committing-Speaker-Future-Action`). But reliability (in terms of inter-annotator agreement) on the latter category has been found to be low ([7] report $\kappa = 0.15$, partly due to the ambiguity in distinguishing `acknowledgements` from `acceptances` – e.g., does "okay" mean "I understand" or "I'll do that"?). The ICSI MRDA annotation schema [8] incorporates a `commit` dialogue act type, but finds them in only 0.24% of utterances in meetings. And to date, most attempts to automatically tag MRDA-based dialog acts concentrate on five general high-level dialogue act classes [9,10], rather than tagging at a level low enough to distinguish `commit` acts from other statements. More importantly for current purposes, though, these commitment acts do not in any case capture the distinction between action items and more general commitments (i.e. commitments to general courses of action or approaches, as well as to specific concrete tasks).

From these studies, it may be productive to surmise that finding action items in meetings involves identifying an interactive process that dialog acts by themselves do not capture. After all, a dialog act corresponds to the illocutionary force of one utterance made by one person. The process of establishing an action item in a meeting, however, is better represented as a type of group action – or *social action* [11] – that is often coordinated by multiple participants over multiple utterances using multiple sign-systems or modalities. That coordinated action entails a public commitment by a specific person or group to be responsible for a specific action to be carried out within a specific timeframe; and that commitment is made in the presence of, and is acknowledged by, others. As we discuss below, this means that detection is best carried out by trying to detect such group actions – here, by looking for multiple complementary utterance types, but potentially also by including information from other modalities.

3 Baseline Experiments

In [1], an initial annotation of action item subdialogues (and topic segmentations) was performed on 65 meetings from the ICSI and ISL meeting corpora [12,13]. In this exercise, action items were defined as tasks that would be entered on a to-do list, and identified simply as sets of utterances with a brief textual description. The two annotators identified a total of 921 and 1267 utterances respectively as belonging to action items, and inter-annotator agreement was rather low ($\kappa = .36$, where κ is the kappa statistic as formulated in [14]).[1] This approach is therefore roughly parallel to that of [5] to email classification, although note that the inter-annotator agreement seems much lower on our discourse data. We therefore performed a similar experiment to examine automatic classification performance. Like [5] we used support vector machines [15] via the classifier *SVMlight* [16]; their full set of features are not available to us as many are text- or email-specific, but we experimented with combinations of words and n-grams. Performance, however, was poor, with precision, recall and f-score all below 0.25 (perhaps unsurprisingly, given the low human inter-annotator agreement).

Table 1. Baseline Classification Performance

Meeting	Number of AI Utterances	Precision	Recall	F-Score
1	22	0.31	0.50	0.38
2	27	0.36	0.33	0.35
3	18	0.28	0.55	0.37
4	15	0.20	0.60	0.30
5	9	0.19	0.67	0.30

Partly to examine the effect on performance of using a smaller, more homogeneous dataset, and partly in order to compare with our later results (see below), we applied this simple flat annotation schema to a separate sequence of 5 short related meetings produced as part of the CALO project. These meetings were simulated according to a given general scenario, but were not scripted. In order to avoid entirely data- or scenario-specific results (and also to provide an acceptable amount of training data), we then added a random selection of 6 ICSI meetings and 1 ISL meeting from [1]'s annotations. We assessed classification performance via a 5-fold validation on each of the CALO meetings; in each case, we trained classifiers on the other 4 meetings in the CALO sequence, plus the fixed ICSI/ISL training selection. Performance is shown in Table 1; these figures were obtained using *SVMlight* with words (unigrams, after text normalization and stemming) as features – we also investigated other discriminative classifier methods, and the use of 2- and 3-grams as features, but no improvements were gained.

[1] Agreement here is calculated simply with regard to the binary classification of utterances as being action-item-related utterances or not; their classification as belonging to *the same* action item has not been tested.

Overall f-score figures do improve, but are still poor; while recall now may be enough to provide useful feedback to a user (over 50% in most cases), precision is low (probably low enough to make this feedback confusing at best). We did obtain higher precision (in some cases over 50%) by using a simple n-gram-based classification method (we trained a trigram language model, and manually set a suitable likelihood threshold), but this was not consistent across all 5 meetings, and came at the cost of much lower recall (c.10%).

4 Hierarchical Annotations

Two problems will be apparent: firstly, the accuracy is lower than desired; secondly, the identification of related utterances does not in itself go very far towards allowing us to identify the action items themselves, and to extract their associated properties (deadline, owner etc.). It became apparent during this phase that the utterances in question form a very heterogeneous class, including some distinct sub-classes which perform different discourse functions and have their own distinct features. Treating these distinct classes as one leads to a classifier which must be too general to give good precision, or too specific to give good recall. Treating them separately and combining the results (along the lines of [17]) might allow better performance. Our next step was therefore to produce an annotation schema which incorporates an explicit distinction between these distinct utterance sub-classes. As discussed above, we decided on the following criteria to determine if an exchange of dialogue specifies an action item:

1. The content of the exchange specifies a concrete future action discussed in the meeting that someone would write down on a to-do list.
2. There is an explicit person or persons who will carry out the action item, and agreement by that person(s) to do so.
3. There is a fairly specific timeframe for when that action is expected to be performed.

These criteria yielded four classes, as shown in Table 2. The first three correspond to the discussion and assignment of the individual properties of the action

Table 2. Utterance Sub-classes

Key	Class	Description
D	`task description`	Utterances containing an explicit description of the task to be carried out.
O	`owner`	Utterances containing an explicit reference to the responsible party.
T	`timeframe`	Utterances containing an explicit reference to the timeframe for completion.
A	`agreement`	Utterances explicitly signalling acceptance or agreement.

Speaker	Utterance	D	O	T	A
CYA	yeah. also, 'cause you said **you were gonna send me an email about how to set up our travel**.	x	x		
HHI	**yeah, I'm gonna send- yeah, I'll send you the email uhm uhm when I go back. send you the email. uhm and you're gonna have to contact him, and they have a travel agency**.	x	x	x	x
CYA	**okay**.				x

Fig. 1. A nice neat example, where most of the work is done by one utterance. Here, the desired task description might be something like "send CYA an email aout setting up travel "; the timeframe "when I go back", and the owner is HHI.

Speaker	Utterance	D	O	T	A
SAQ	not really. the there was the uh notion of **the preliminary patent**, that uh	x			
FDH	yeah, it is a cheap patent.				
SAQ	yeah.				
CYA	okay.				
SAQ	which is				
FDH	so, it is only seventy five dollars.				
SAQ	and it is it is e an e				
CYA	hm, that is good.				
HHI	talk to				
SAQ	yeah and and it is really broad, you don't really have to define it as w as much as in in a you know, a uh				
FDH	yeah.				
HHI	I actually think **we should apply for that right away**.	x	x	x	
CYA	**yeah, I think that is a good idea**.				x
HHI	I think **you should, I mean, like, this week**, s start moving in that direction. just 'cause that is actually good to say, when you present your product to the it gives you some instant credibility.		x	x	
SAQ	[Noise]				
SAQ	**mhm**.				x
CYA	**right**.				x

Fig. 2. A messier example. Here the desired task description might be something like "apply for preliminary patent", and so the `description` utterances must include SAQ's original mention of "preliminary patent" as well as HHI's proposal to "apply for" it. The timeframe would be "right away, this week", and the owner seems to be "you" (whoever HHI is addressing).

item: the associated `task description`, the `timeframe` for completion of that task, and the `owner` or party responsible for it. The final class is `agreement`, which covers utterances which explicitly show that the action item is accepted or agreed upon. The classes are shown in Table 2, and annotation examples in Figs. 1 and 2.

Speaker	Utterance	D	O	T	A
CYA	so, **both the charger and the interface need to be designed**, 'cause **you need** to figure out how **you're gonna** attach the charger to the batteries.	x	x		
SAQ	**yeah**, [Smack] **yeah**. and then that that is where **we'll** address the issue of the parallel versus series of configuration of the batteries.		x		x
AOF	**yeah, good call**. [Noise]				x
CYA	**right**.				x
CYA	I think some of these things obviously we wanna get as much done before the meeting as possible, but some of them can uh will have to wait, like marketing will have to wait. the testing can wait. we don't really need to get into that yet. and the battery charger is something that				
FDH	mhm.				
HHI	we don't need to do that anyway.				
CYA	exactly. because we can just put fresh batteries in right now, if we need to.				

Fig. 3. An example of a less concrete task decision. Here, there is a decision, a joint commitment that there is a task which needs doing ("design the charger and interface", perhaps), and it seems to get agreed - so a conscientious note-taker might add that to a to-do list. But there is no concrete assignment to a person, and no definite time frame (in fact, they seem to decide NOT to take action on this task immediately, but leave it till later).

More specifically, annotation with the `task description` subclass includes any utterances that specify what action is to be done, including the utterances that provide required antecedents for anaphoric references: as "notion of the preliminary patent" does for the statement "we should apply for that right away" in Fig. 2. In short, it includes any utterances that contain the actual words that would be used to put together a short description of the task.

Annotation with the `owner` subclass includes any utterances that explicitly specify who is responsible for ensuring that the action is carried out, as with "you should ...start moving in that direction", but not e.g. those whose function (rather than explicit surface form) might be taken to do so implicitly (such as agreements by the reponsible party). The `timeframe` subclass includes any utterances that explicitly refer to when a task may start or when it is expected to be finished; this is often not specified with an exact date – as with, "by the end of next week," or "before the trip to Aruba" – but the time that the action is expected to be performed should still be fairly clear.

Finally, the `agreement` subclass includes any utterances in which people agree that the action should and will be done. These are often acknowledgements by the owner to carry out the task, but can be utterances made when other people express their agreement that an action should be done or that a particular person should do it.

Note that a single utterance may be assigned to more than one of these classes: "**John, you** should do that **by next Monday**" might count as `owner` and

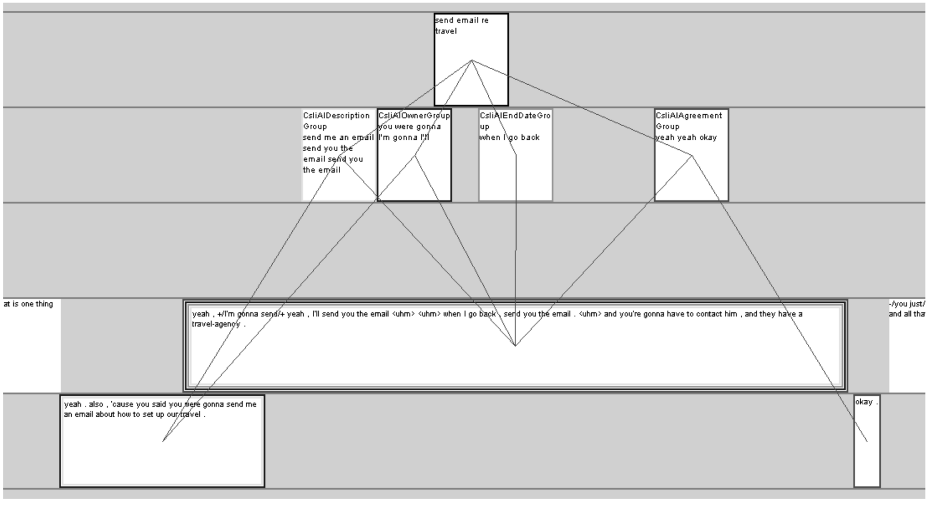


Fig. 4. An example of the hierarchical annotation scheme for a single action item

`timeframe`. Likewise, there may be more than one utterance of each subclass for a single action item: John's response "OK, **I'll** do that" would also be classed as `owner` (as well as `agreement`).

While we are not currently requiring all of these subclasses to be present for a set of utterances to qualify as denoting an action item, we expect any action item to include most of them. Figure 3 shows an example of a less concrete task decision, *not* classified as an action item in this case. As we annotate more data, we hope to get a more concrete idea of the effect of the presence/absence of the individual classes, and hope to use that to gain more insight into the distinction between the specific action items we concentrate on here, and more general decisions and commitments.

To date, we have applied this annotation schema to 12 meetings, using the NOMOS annotation software [18]: the sequence of 5 related short CALO meetings, and a random selection of 6 ICSI meetings and 1 ISL meeting (as described in the baseline section above). An initial assessment of reliability between 2 annotators on the single ISL meeting (chosen as it presented a significantly more complex set of action items and annotations than the others in this current set) was encouraging, with reasonable figures for the kappa statistic for each of the distinct sub-classes. The best agreement was achieved on `timeframe` utterances ($\kappa = .86$), with `owner` utterances slightly less good (between $\kappa = .77$), and `agreement` and `description` utterances worse but still acceptable ($\kappa = .73$). Further annotation and refinement of the schema is in progress.

5 Experiments

Given the small amount of data currently available, a full evaluation of the proposed classification and detection approach is not possible, but we can perhaps

get some indications. We first trained individual classifiers for each of the utterance sub-classes. For `agreement` utterances, we used a naive n-gram classifier very similar to that of [10] for dialogue act detection, scoring utterances via a set of most predictive n-grams of length 1–3 and making a classification decision by comparing the maximum score to a threshold (where the n-grams, their scores and the threshold are automatically extracted from the training data). For `owner`, `timeframe` and `task description` utterances, we used support vector machines as before, with word unigrams as the features (2- and 3-grams gave no improvement – we expect that this is due to the small amount of training data). Again, we cross-validated by testing on each of the 5 CALO meetings separately, with the training set in each case being the other 4 CALO meetings, plus the fixed ICSI/ISL set. Performance varied greatly by sub-class (see Table reftab:res1), with some (e.g. agreement) achieving higher accuracy than the baseline flat classifications, but others being worse. As there is now significantly less training data available to each sub-class than there was for all utterances grouped together in the baseline experiment, worse performance might indeed be expected; it is encouraging that some sub-classes do better. The worst performing class is `owner`; we suspect parse information may help here (see below).

Table 3. Sub-class Classification Performance

Class	Precision	Recall	F-Score
D	0.23	0.41	0.29
O	0.12	0.28	0.17
T	0.19	0.38	0.26
A	0.48	0.44	0.40

However, even with poor performance for some of the individual sub-classifiers, we should still be able to combine them to get a benefit as long as their true positives correlate better than their false positives (intuitively, if they make mistakes in different places). So far we have only conducted an initial naive experiment, in which we combine the individual classifier decisions in a weighted sum over a window (currently set to 5 utterances). If the sum over the window reaches a given threshold, we hypothesize an action item, and take the highest-confidence utterance given by each sub-classifier in that window to provide the corresponding property. As shown in Table 4, this gives reasonable performance on most meetings, although it does badly on meeting 5 (apparently because no explicit agreement takes place, while our manual weights emphasised agreement).[2] Most encouragingly, the correct examples provide some useful "best" sub-class utterances, from which the relevant properties could be extracted – see Fig. 5.

We are confident that these results can be improved significantly: rather than sum over the binary classification outputs of each classifier, we can use their

[2] Accuracy here is currently assessed only over correct detection of an action item in a window, not correct assignment of all sub-classes.

Table 4. A first experiment at combined classification

Meeting	Number of AIs	Correct	False Pos	False Neg	F-Score
1	3	2	1	1	0.67
2	4	1	0	3	0.40
3	5	2	1	3	0.50
4	4	4	0	0	1.00
5	3	0	1	3	0.00

T the start of week three just to
O reconfirm everything and at that time jack i'd like you to come back to me with the
D the details on the printer and server
A okay

O so jack uh for i'd like you to
D have one more meeting on um uh
T in in a couple days about uh
A okay

Fig. 5. Examples from meeting 4, with "best" sub-class utterances in dialogue order

confidence scores or posterior probabilities, and learn the combination weights to give a more robust approach. There is still a long way to go to evaluate this approach over more data, and to evaluate the accuracy and utility of the resulting sub-class utterance hypotheses.

6 Discussion and Future Work

We have shown that taking a notion of structure into account seems advantageous when detecting action items in spoken discourse. Without one, classification accuracy is limited; with one, we believe that accuracy can be improved, and the detected utterances can be used to provide the properties of the action item itself. An interesting question is how and whether the notion of structure we use here relates to notions of discourse structure in more general use. If a relation exists, this would help shed light on the decision-making process we are attempting to (begin to) model; and might provide us with a way of using other more plentiful annotated data.

The main priority for our current and future efforts is the annotation of more meetings in order to obtain sufficiently large training and test sets. This effort will concentrate on those meetings from the ICSI, ISL, and CALO corpora which contain decision-making dialogues (in some types of meeting, action items are very sparse). Once more annotated data is available, we will also be able to examine the CALO and ICSI corpora for correlations with existing annotations for other related phenomena, such as meeting acts [19] and dialog acts [8], which may add useful information for features not currently being modelled.

Another priority is to examine the effect on performance when working from speech recognition hypotheses (as opposed to the human transcripts used in this

paper), and the best way to incorporate multiple hypotheses (either as n-best lists or word confusion networks). This will allow us to incorporate action item detection into a working system, e.g. the CALO assistant.

We are also actively investigating alternative approaches to sub-classifier combination: the method used so far is rather ad-hoc and manually defined by trial and error, and better performance (and a more robust and trainable overall system) might be obtained by using a Bayesian network, or a maximum entropy classifier as used by [17].

Another avenue of research we will be pursuing in collaboration with vision and speech researchers on the CALO project will be to integrate multimodal and paralinguistic information as model features. In particular, we expect gaze, head pose, and prosody to help in distinguishing action item agreement and assignment utterances from less relevant classes (e.g. backchannels); and we are examining the incorporation of written and drawn information (in particular, milestones drawn on project sketches) to improve deadline extraction.

We are also developing an interface to a new large-vocabulary version of the Gemini parser [20], allowing us to use semantic parse information, firstly as features in the individual sub-class classifiers, and secondly to extract entity and event representations from the classified timeframe, owner and task description utterances – eventually working towards a full semantic representation for the action item [21]. This can then be supplied to other agents within the CALO system to provide useful functionality for a user, such as the automatic addition of entries to calendars and to-do lists.

Acknowledgements

This work was supported by the CALO project, DARPA grant NBCH-D-03-0010. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

1. Gruenstein, A., Niekrasz, J., Purver, M.: Meeting structure annotation: Data and tools. In: Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue. (2005) 117–127
2. Lisowska, A., Popescu-Belis, A., Armstrong, S.: User query analysis for the specification and evaluation of a dialogue processing and retrieval system. In: Proceedings of the 4th International Conference on Language Resources and Evaluation. (2004)
3. Banerjee, S., Rosé, C., Rudnicky, A.: The necessity of a meeting recording and playback system, and the benefit of topic-level annotations to meeting browsing. In: Proceedings of the 10th International Conference on Human-Computer Interaction. (2005) 643–656
4. Nicholas C. Romano, J., Jay F. Nunamaker, J.: Meeting analysis: Findings from research and practice. In: Proceedings of the 34th Hawaii International Conference on System Sciences. (2001)

5. Corston-Oliver, S., Ringger, E., Gamon, M., Campbell, R.: Task-focused summarization of email. In: Proceedings of the Text Summarization Branches Out ACL Workshop. (2004)
6. Allen, J., Core, M.: Draft of DAMSL: Dialog act markup in several layers (1997) http://www.cs.rochester.edu/research/cisd/resources/damsl/RevisedManual/.
7. Core, M., Allen, J.: Coding dialogues with the DAMSL annotation scheme. In Traum, D., ed.: AAAI Fall Symposium on Communicative Action in Humans and Machines. (1997) 28–35
8. Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., Carvey, H.: The ICSI Meeting Recorder Dialog Act Corpus. In: Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue. (2004) 97–100
9. Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Ess-Dykema, C.V., Martin, R., Meteer, M.: Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational Linguistics **26**(3) (2000) 339–373
10. Webb, N., Hepple, M., Wilks, Y.: Dialogue act classification using intra-utterance features. In: Proceedings of the AAAI Workshop on Spoken Language Understanding. (2005)
11. Goodwin, C.: Action and embodiment within situated human interaction. Journal of Pragmatics **32** (2000) 1489–1522
12. Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., Wooters, C.: The ICSI Meeting Corpus. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). (2003) 364–367
13. Burger, S., MacLaren, V., Yu, H.: The ISL Meeting Corpus: The impact of meeting type on speech style. In: Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2002). (2002)
14. Siegel, S., N. J. Castellan, J.: Nonparametric Statistics for the Behavioral Sciences. 2nd edn. McGraw-Hill (1988)
15. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer (1995)
16. Joachims, T.: Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods – Support Vector Learning. MIT Press (1999)
17. Klein, D., Toutanova, K., Ilhan, H.T., Kamvar, S.D., Manning, C.D.: Combining heterogeneous classifiers for word-sense disambiguation. In: Proceedings of the ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions. (2002) 74–80
18. Niekrasz, J., Gruenstein, A.: A software framework for semantic annotation of multimodal discourse. In: The 5th Conference on Language Resources and Evaluation (LREC '06). (2006)
19. Bates, R., Menning, P., Willingham, E., Kuyper, C.: Meeting acts: A labeling system for group interaction in meetings. In: The 9th European Conference on Speech Communication and Technology (Interspeech '05). (2005) 1589–1592
20. Dowding, J., Gawron, J.M., Appelt, D., Bear, J., Cherny, L., Moore, R., Moran, D.: Gemini: A natural language system for spoken language understanding. In: Proc. 31st Annual Meeting of the Association for Computational Linguistics. (1993)
21. Niekrasz, J., Purver, M., Dowding, J., Peters, S.: Ontology-based discourse understanding for a persistent meeting assistant. In: Persistent Assistants: Living and Working with AI: Papers from the 2005 AAAI Spring Symposium. (2005)

Overlap in Meetings: ASR Effects and Analysis by Dialog Factors, Speakers, and Collection Site

Özgür Çetin[1] and Elizabeth Shriberg[1,2]

[1] International Computer Science Institute, Berkeley, CA, USA
[2] SRI International, Menlo Park, CA, USA
{ocetin,ees}@icsi.berkeley.edu

Abstract. We analyze speaker overlap in multiparty meetings both in terms of automatic speech recognition (ASR) performance, and in terms of distribution of overlap with respect to various factors (collection site, speakers, dialog acts, and hot spots). Unlike most previous work on overlap or crosstalk, our ASR error analysis uses an approach that allows comparison of the same foreground speech with and without naturally occurring overlap, using a state-of-the-art meeting recognition system. We examine a total of 101 meetings. For analysis of ASR, we use 26 meetings from the NIST meeting transcription evaluations, and discover a number of interesting phenomena. First, overlaps tend to occur at high-perplexity regions in the foreground talker's speech. Second, overlap regions tend to have higher perplexity than those in nonoverlaps, if trigrams or 4-grams are used, but unigram perplexity within overlaps is considerably lower than that of nonoverlaps. Third, word error rate (WER) after overlaps is consistently lower than that before the overlap, apparently because the foreground speaker reduces perplexity shortly after being overlapped. These appear to be robust findings, because they hold in general across meetings from different collection sites, even though meeting style and absolute rates of overlap vary by site. Further analyses of overlap with respect to speakers and meeting content were conducted on a set of 75 additional meetings collected and annotated at ICSI. These analyses reveal interesting relationships between overlap and dialog acts, as well as between overlap and "hot spots" (points of increased participant involvement). Finally, results from this larger data set show that individual speakers have widely varying rates of being overlapped.

1 Introduction

Speaker overlap is frequent in natural conversation For example, in the 26 different meetings we look at in this work (from the last four years' NIST meeting speech recognition evaluations) an average of roughly 12% of all foreground speaking time is overlapped by speech from one or more other talkers. The ratio is even higher (30 to 50%) if one considers pause-delimited regions as units, rather than speaking time [17].

While general effects of overlap are well reported in the literature (e.g., [3], [10], [12], [16], [17], and [21]), there is relatively little work quantifying such

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 212–224, 2006.

effects under the different conditions we consider. In particular, to the best of our knowledge, the issue of the effect of overlaps on ASR errors *adjacent* to overlap regions has received little attention in earlier work. We explore this question by using a method that allows us to compare ASR results for the same foreground speech with and without naturally occurring overlap. For data from the six different meeting collection sites, we examine both ASR and language model perplexity as a function of the presence or absence of crosstalk, its severity, and time distance before and after the overlap.

To begin to better understand patterns of overlap with respect to meeting content, we use an additional set of 75 ICSI meetings that are independently hand-annotated for dialog acts and hot spots. We ask whether overlap is associated with specific dialog acts, and in turn whether such information can shed light on perplexity patterns and ASR results. We also ask to what degree hot spots are correlated with overlap, since increased involvement would be assumed to predict increased overlap. Finally, since the ICSI data set contains significant amounts of data per speaker, we ask how individual speakers vary in terms of how frequently they are overlapped by another talker.

2 Method

2.1 Data

For analyses of ASR performance, we use 19.8 hours of recordings from 26 different meetings from the 2002, 2004, and 2005 NIST meeting speech recognition evaluations [11]. These meetings were provided by the sites AMI (2), CMU (6), ICSI (6), LDC (4), NIST (6), and VT (2), with the number of meetings given in parentheses. The number of participants varies from three to nine, and the total amount of speech in the individual headset microphones (IHMs) after segmentation is about 3.5 hours. For further analyses, requiring human annotations, we use a set of 75 meetings from the ICSI meeting corpus [9]. In separate efforts, this set was extensively hand-marked for dialog acts [6] as well as for hot spots [20]. Since this data was included in training the ASR system, we did not use it in analyses of recognition or perplexity.

2.2 Recognition System

Recognition experiments are conducted using the 2005 ICSI-SRI meeting system [19]. This system is adapted from SRI's conversational telephone speech system to the meeting domain using a variety of meeting data, excluding the test data. The standard n-gram language models (LMs) with order as high as four were trained on standard text and meeting transcriptions as well as on Web texts [4]. We use manual reference segmentations in our experiments and analyses to avoid confounds with the automatic speech segmentation errors.

2.3 Experiment Conditions

We use synchronously recorded speech from IHMs and speech/nonspeech alignments to create a rendition of crosstalk that is accurate in terms of speech that has overlapped and crosstalk severity. First, each channel is normalized to have unit energy using the average energy of speech samples in that channel. Next, to each channel the remaining channels are added in a time-synchronous fashion, after an appropriate linear weighting to adjust crosstalk severity (referred to as the "crosstalk condition"). To provide a contrast condition for isolating effects of background noises, we perform a second set of experiments, where a channel from the remaining channels is added only if no speech activity is marked for that channel (referred to as the "background-noise condition"). The performance differences between the crosstalk and background-noise conditions should indicate the crosstalk effects mainly due to the actual speech as opposed to background noise. See Figure 1 for an illustration of the design.

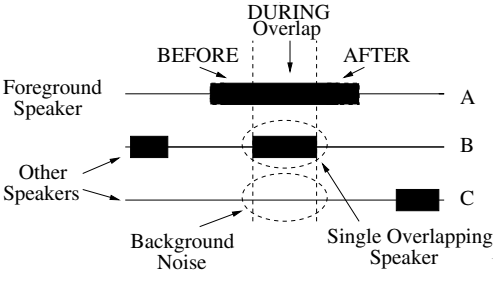


Fig. 1. Illustration of experiment conditions. When A is taken as the foreground speaker, B and C are background speakers. For the crosstalk condition, the original audio from B and C is added to A. For the background-noise condition, B and C are added only in the cases in which they do not contain any speech (e.g., inside DURING, B is not added to A and only C is added). BEFORE and AFTER in A are nonoverlaps. Solid rectangles denote speech segments.

It is important to note that the crosstalk condition contains only speech that actually occurred at the same time. We do not create crosstalk using speech from different corpora or time spans. Nevertheless, the waveform addition is admittedly simplistic and does not capture some aspects of crosstalk such as nonlinear frequency weighting and reverberation. However, the effects from these factors would act only to exacerbate effects we report here. Our study uses the performance difference between results with and without crosstalk in the same region of speech, and at this level of relative comparison such effects would be roughly normalized out. Also, to assess generalizability we repeat our crosstalk experiments with mixing powers 1/4, 1/2, and 1.

3 Results

3.1 Rate of Overlap by Site

Table 1 provides rates of overlap in the evaluation test data from the six different sites, along with the rate overall. Rates are computed as the ratio of the time during which a foreground talker is speaking while overlapped, to the total amount of foreground speaking time over all foreground talkers. As shown, four sites have rates ranging from 11% to 13.3%, which is quite close considering that the meetings are of different natures. Two sites, AMI and VT, have significantly lower rates; this suggests that these two meeting types may be more artificial in terms of interaction patterns. For all sites but VT, over 90% of the overlaps involve only one background speaker, even though the meetings involved more than two speakers. VT shows a somewhat different pattern, with a higher rate of multiple-speaker overlaps, and yet a lower rate of overlap overall. This suggests that in VT meetings, overlap may be associated with a different function than it is in the five other meeting types.

Table 1. Rates (%) of overlap by site. Line 1 provides the percentage of speech duration that is overlapped by any number of speakers. Line 2 considers all overlap events, and provides the percentage of overlaps that involve only two speakers.

Rate	*All*	*AMI*	*CMU*	*ICSI*	*LDC*	*NIST*	*VT*
total overlap time/total speaking time	11.6	7.1	13.3	13.0	12.3	11.0	6.1
single-speaker overlap time/total overlap time	92.2	93.0	91.0	91.0	94.3	92.7	85.2

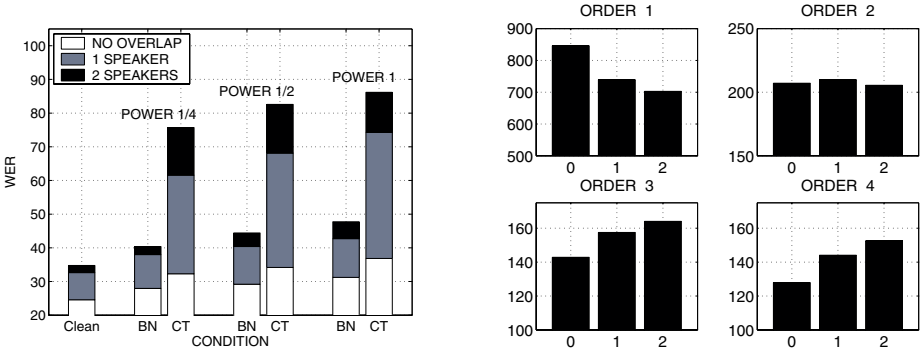
3.2 ASR and Perplexity by Overlap Condition

WERs for various recognition conditions are provided in Table 2. WERs in this table are cumulative for all segments of the test data; analyses for overlaps and nonoverlaps are provided later. Expectedly, both the crosstalk and background noise significantly degrade recognition performance (up to 60% relative) and the degradation is more severe in the crosstalk condition. For future reference, we note that the 4-gram perplexity of all the meetings is 131, 111 for AMI, 148 for CMU, 102 for ICSI, 133 for LDC, 143 for NIST, and 188 for VT meetings.

Using the time marks of the reference transcriptions obtained from a forced alignment and time marks in the recognition output, we found errors in the nonoverlap regions, and in the single- and two-speaker overlap regions [5]. WERs for each region type were calculated from the number of substitutions, insertions, deletions, and reference words assigned to the regions of that type. WERs for each recognition condition across overlap/nonoverlap types are displayed in Figure 2(a). We discover that crosstalk significantly increases WER, much more so than does background noise, and that two-speaker overlaps cause more errors than single-speaker overlaps. The pattern of results was similar across different sites (not shown here due to space restrictions).

Table 2. WERs (%) under different recognition conditions. Clean refers to the case when the original IHM audio is used, and crosstalk and background are the crosstalk and background-noise conditions, respectively (cf. Figure 1). Mixing power is the square of the linear mixing coefficient for the interfering channels, assuming a coefficient of 1 for the channel with interference.

Condition	*Mixing Power*	*All*	*AMI*	*CMU*	*ICSI*	*LDC*	*NIST*	*VT*
Clean	N/A	25.6	19.1	28.0	16.2	29.0	22.8	20.6
Background	1/4	29.1	22.5	31.2	20.7	32.1	26.2	23.3
Crosstalk	1/4	36.4	26.1	41.0	29.5	42.3	35.1	26.5
Background	1/2	30.6	23.4	32.8	22.6	33.1	27.8	24.4
Crosstalk	1/2	38.8	28.2	43.9	33.4	44.5	38.1	28.0
Background	1	32.6	24.6	34.9	25.6	34.6	30.2	24.8
Crosstalk	1	41.7	30.7	47.2	37.7	47.7	41.9	29.1



(a) WER (b) Perplexity

Fig. 2. (a) WERs for the clean, background noise (BN), and crosstalk (CT) conditions with the mixing powers 1/4, 1/2, and 1. For each condition, we display the WER in a stacked fashion for nonoverlaps, and single-speaker and two-speaker overlaps. (b) Perplexities of the foreground reference words during nonoverlaps (0), single-speaker overlaps (1), and two-speaker overlaps (2), for various n-gram LMs.

Perplexities for the nonoverlap and single- and two-speaker overlap regions are displayed in Figure 2(b). The perplexities here are those of the reference words corresponding to these regions in the foreground speaker's speech, since we would like to find out whether the speech from overlaps or nonoverlaps could be inherently more difficult to predict lexically. As shown in Figure 2(b), there is a reversal of the relationship between perplexity and the number of simultaneous speakers. Overlap regions tend to have higher perplexity than those in nonoverlaps if trigrams or 4-grams are used, but the unigram perplexity within overlaps is considerably lower than that of nonoverlaps. (While the perplexities were aggregated over the different sites, individual sites show a similar overall pattern, suggesting robustness of the results.)

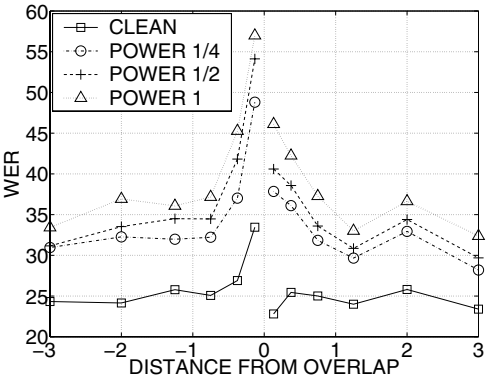
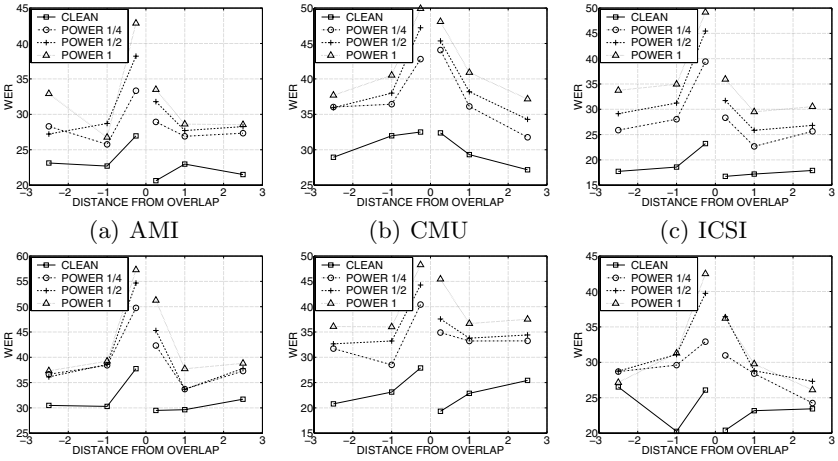


Fig. 3. WERs (%) for clean and crosstalk conditions with various gains, as a function of distance from the overlap (in seconds). Negative distances correspond to before overlaps, and positive distances to after overlaps. Note that the WER at each point represents data coming from that distance only (i.e., results are not cumulative).

An analysis of the frequent n-grams in the test data provided some insight. We found that overlaps contained far more backchannels and discourse markers than nonoverlaps, and the degree of increase for both types of events was larger when the number of simultaneous speakers was higher. Because backchannels are frequent unigrams in LMs trained on spontaneous speech, unigram perplexity is lower when the number of overlapping talkers is higher. The longer n-grams in nonoverlap regions tend to be within-sentence sequences, such as *might be able to* and *just a matter of*, which are relatively common in ASR LMs. But, in overlap regions, we see far more cases like *right right right so* and *right i i am*, which are frequent at turn exchanges but not in ASR LMs, since most n-gram tokens come from regions inside single-speaker turns in which the speaker has already obtained the floor. In Section 3.3, we will provide a more detailed quantification of the relationship between the speaker overlaps and dialog acts.

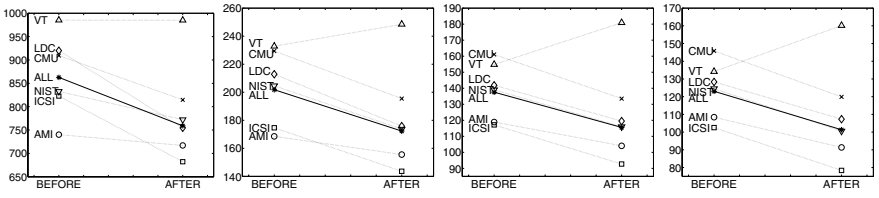
Using the reference and recognition time marks, we looked at recognition errors associated with the nonoverlap regions directly before and after an overlap. We restricted the analysis to the errors that were completely included within such nonoverlap regions, in order to avoid any potential acoustic bias in error rates from the overlap region itself. In Figure 3, we plot WER over such before- and after-overlap regions for different recognition conditions, as a function of the distance from the overlap. As shown, WER decreases as a function of distance from the overlap. In addition, there is an asymmetry in the errors before and after overlaps: WERs are higher before the overlap than after it. This finding is consistent across different recognition conditions and across meetings from different sources (cf. Figure 4) and unlikely to be due to the recognizer itself because the decoding is not strictly forward in time, and not due to reverberation because its effects are smaller, for example, less than 250 ms in the ICSI meetings.



(a) AMI (b) CMU (c) ICSI

(d) LDC (e) NIST (f) VT

Fig. 4. WERs (%) as a function of time surrounding overlaps for various sites



(a) 1-gram (b) 2-gram (c) 3-gram (d) 4-gram

Fig. 5. Perplexities of the foreground reference words before and after overlaps with respect to various n-gram LMs

To investigate whether the lower error rate just after the overlap can be attributed to lexical effects, we calculated perplexities of the reference words in these regions (cf. Figure 5). For all n-gram orders, perplexity is lower after overlaps than before them, and the same general pattern holds for each of the different meeting collection sites except VT (cf. Section 2.1) so it appears to be a robust finding. Although further investigation is needed, we hypothesize from inspection of the frequent n-grams that the lower perplexity after overlaps stems at least in part from a tendency to begin new sentences at this location.

3.3 Dialog Acts and Speaker Overlap

To further understand the pattern of results, which so far have treated all speech as one class, we look at various breakdowns of the speech. A first interesting breakdown is by basic dialog act, for example, whether an utterance is a statement, question, backchannel, or other type. As mentioned in Section 3.2, overlaps

Table 3. Dialog acts in the ICSI meeting corpus and their examples

Dialog Act	*Example*
Backchannel	*right, uhhuh*
Disruption	*so that's* – (self- or other-interruption)
Floor grabber	*well i* - (context suggests trying to gain floor)
Question	*and the new machines are faster?*
Statement	*it's easier just to buy new disks.*
Unlabelable	Unintelligible or some other issue

contain far more backchannels and discourse markers than nonoverlaps, and the degree of increase for both types of events was larger when the number of simultaneous speakers was higher. Here we produce the complete statistics on which speech regions in an overlap analysis are associated with which acts.

Fortunately, we can investigate the relationships between speaker overlaps and dialog acts by using the ICSI MRDA corpus [18], which contains hand-annotations for dialog acts [6] and hot spots [20] for the 75 meetings in the ICSI meeting corpus [9]. These meetings were recorded as part of the same data collection effort, and are similar in style and content to the meetings in the corpus. Roughly 16% of all speech in the annotated ICSI meeting corpus is overlapped, which is close to the value of 13% found for the unannotated ICSI evaluation data. The higher rate in the annotated corpus is most likely due to the subtype of ICSI meetings in each set, with the annotated set containing many meetings involving familiar participants who met regularly.

Because we want to know what dialog acts speakers actually produced, we look at human annotations based on reference transcripts. Dialog acts were labeled in detail [6], but collapsed into five broad classes for purposes of these analyses, which are listed in alphabetical order in Table 3. These basic classes have been used in much of the work on automatic detection of dialog acts for this data [2], [1], [8], [22]. Important for these analyses is that the annotation of dialog acts themselves does not depend explicitly on acoustic overlap [6]. For example, a backchannel, such as *uh-huh* can occur either during or after another speaker's contribution. Similarly, a disruption (uncompleted utterance) can be disrupted by the same or a different speaker. A floor grabber (attempt to gain the floor) can occur during or outside of other speaker's speech, and is labeled regardless of whether or not the floor is actually obtained.

We use time measures from a forced alignment of the reference transcriptions in the analyses to follow, because the average length of words in a dialog depends on the dialog act (e.g., words in backchannels or floor grabbers tend to be shorter than words in statements or questions). Overall, an average of just over 16% of 53.5 hours of speaking time on a foreground channel is overlapped by one or more other talkers. If we break down this 16% to see what it is made up of in terms of dialog acts, we find that there is a clear association between certain acts and

Table 4. Columns 2-4 display expected versus observed percentages of in-dialog-act times within the 16% of total speaker time that is overlapped. Expected values are based on the distribution of in-dialog-act times for the overall corpus. Columns 5–7 display expected versus observed percentages of overlap time, given a dialog-act class. Expected values are the rate of overlap in the overall corpus. Relative difference percentages are those of the observed values from the expected values.

Dialog Act	*In-Dialog-Act Time*			*Overlap Time*		
	Expected	*Observed*	*Rel. Diff.*	*Expected*	*Observed*	*Rel. Diff.*
Backchannel	4.9	13.7	+179.6	16.0	69.5	+333.3
Disruption	12.8	15.7	+22.7	16.0	19.5	+48.0
Floor grabber	1.5	3.8	+153.3	16.0	19.5	+21.7
Question	7.3	5.9	– 19.2	16.0	15.2	– 5.0
Statement	71.5	58.6	– 18.0	16.0	12.4	– 22.5
Unlabelable	1.9	2.3	+21.1	16.0	28.8	+79.4

overlap. Table 4 shows expected versus observed results for in-dialog-act times during overlap, and the rate of overlap from the perspective of dialog acts.

We observe in Table 4 that backchannels and fillers are much more likely to occur within overlap regions than would be expected from their distribution overall in the corpus. Disruptions and unlabelable utterances also occur more than expected. The longer, propositional-content-based utterances, questions and statements, are relatively less likely during overlap. Note that the large relative increase for backchannels and fillers is balanced out by a smaller relative increase in statements and questions, because the latter types have more words (and longer words) than the other utterance types. As explained earlier, the hand-coding of dialog acts was not based on whether or not an utterance occurred during overlap. Thus, the biases shown in Table 4 are not predetermined by the hand labels for the dialog acts. Rather, they reflect an association between certain dialog act types in foreground speech when the talker is overlapped, and the functions of these utterances in the meeting.

We can see in Table 4 that the most dramatic act for predicting overlap is the backchannel: If a foreground talker is producing a backchannel, the probability that he is being overlapped by one or more talkers is nearly 70%. Disruptions and unlabelable utterances are the next highest conditional predictors of overlap. One very interesting observation is that floor grabbers are only about 20% more likely to be uttered during overlap than expected. This suggests that when speakers try to grab the floor, they may be trying to do so during silent regions in the other talkers' speech. The probability of overlap is lowest during statements and questions, suggesting that much of the overlap is not blatant interruption of propositional content, but rather occurs at potential turn-exchange regions in the discourse. This is consistent with long-standing work in conversation analysis [13], [14], [7], [15] but to our knowledge has not previously been analyzed using close study of acoustic overlaps in a large corpus of meeting data.

3.4 Hot Spots and Speaker Overlap

We were also interested in the relationship between overlap and hot spots, or locations in the meetings in which speakers become more affectively involved. The ICSI meeting corpus is hand-labeled for such hot spots, using a procedure described in [20]. Each hot spot consists of one or more utterances across different speakers, and has a number of internal structural and categorical markings (such as start, end, local peaks in hotness, level of hotness, and type of hotness). For purposes of this work, such codings were collapsed, and we asked simply whether an utterance was part of versus not part of a hot spot. Labeling of hot spots tried to capture speaker-normalized animation within utterances, rather than the rate of utterance exchanges. Starts and ends of hot spots were determined by semantic content, but their status as a hot spot relied on individual emotionally salient utterances within a talker. Hot spots were allowed to occur within only one talker's speech, but in general we assumed that the animation of one speaker tended to produce more interaction with other talkers.

Table 5 shows that there is indeed an association between hot spots and overlap. As shown (see the expected column for hot spots in the corpus overall, under line 2 of the table) hot spots themselves are fairly rare overall in the data, occurring during less than 5% of speaking time. If we look only at overlap regions, hot spots are about 50% more probable. This means that there are many remaining hot spots whose overlap patterns match those of the overall corpus; the "hotness" in these cases must come from aspects of the individual speakers' utterances. Conversely, many overlap regions contain utterances that are not hot, since the 16% rate of overlap for the corpus increases to only 25% when conditioned on utterances in hot spots. Thus, while there is a significant association between hot spots and overlap, they appear to reflect distinct phenomena.

Table 5. Expected versus observed values for association between overlap and hot spots. Expected values are the overall rate (%) of overlap (line 1) and the overall rate (%) of hot spots (line 2) in the corpus.

Rate of	*Given*	*Expected*	*Observed*	*Rel. Diff.*
Overlap	Hot spot	16.0	25.2	+57.5
Hot spot	Overlap	4.8	7.5	+36.0

3.5 Overlap Rates by Speaker

As a final analysis, we looked at rates of overlap associated with individual speakers. These rates reflect the proportion of time that one or more other talkers overlap with the foreground talker, given that the foreground talker is speaking. We analyzed 52 speakers; the average amount of data per speaker was about an hour, 10 hours for a speaker with the most data. Results are shown in Figure 6. We discover that there is a very large range of behaviors from different talkers. While many speakers cluster near the 16% overlap value for the corpus

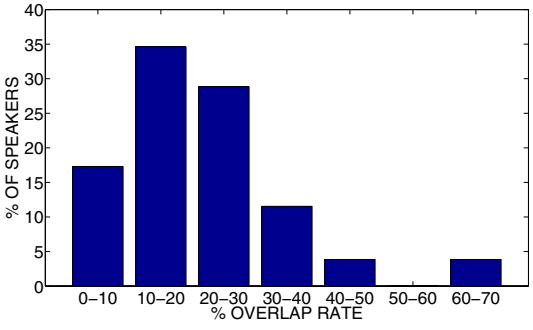


Fig. 6. Rates (%) at which an individual foreground talker is overlapped

overall, 20% of the talkers are overlapped by others more than 30% of the time—with two speakers overlapped between 60 and 70% of the time. Such speakers may be producing only backchannels most of the time, or they may be trying to grab the floor while others are talking and not succeeding.

4 Summary and Conclusion

We analyzed overlaps with respect to ASR performance and language model perplexity in regions before, during, and after the overlap. Using an approach that allowed us to compare the same actually overlapped foreground speech with "clean" and "background-noise" versions, we assessed the relative detriment to ASR of overlapping speech under different crosstalk gain conditions. We found that overlap tends to start at times during which the foreground talker is producing relatively high perplexity word sequences, and that the relationship between perplexity and number of simultaneous talkers is positive for longer n-grams, but negative for unigrams. We discovered a robust asymmetry in ASR error rates before versus after overlaps, apparent across data from the different collection sites. The asymmetry suggests that after being overlapped, the foreground talker temporarily drops to lower-perplexity word sequences, often recycling such events before continuing to talk.

Analyses of a large amount of hand-labeled ICSI meeting data explored the relationship between overlap and content in meetings. Independent dialog act annotations, which did not use overlap as a labeling criterion, showed strong associations with overlap regions. Consistent with classical literature in conversation analysis, but to our knowledge not shown in an automatic analysis of large amounts of meeting data, dialog acts that manage interaction (backchannels, floor grabbers, and disruptions) were positively correlated with overlap, while dialog acts pertaining to propositional content (questions and statements) were negatively correlated. Overlap was also positively correlated with hot spots, or regions of high involvement. Many hot spots, however, showed default rates of overlap, indicating that speaker involvement ratings are based not only on

turn-taking patterns but also on aspects of individual utterances. Finally, individual speakers varied widely in rates of being overlapped; a significant number of speakers showed rates over 30%, with some showing rates over 60%.

Overall, we hope these results illustrate that overlap is an inherent property of natural conversation, and that it shows systematic relationships with word sequences both during and surrounding the overlap. The correlations with word sequences reflect associations at the level of dialog acts, which serve different functions in interaction, as well as at the higher level of hot spots, or greater participant effect. From the engineering perspective, these associations show up as differences in perplexity and WER. Such differences suggest that we may benefit from more intelligent models of overlap in automatic meeting understanding.

Acknowledgments. This work is supported in part by AMI (FP6-506811) and CALO (NBCHD-030010) funding at ICSI and SRI, respectively. The opinions and conclusions are those of the authors and not necessarily endorsed by the sponsors.

References

1. J. Ang, Y. Liu, and E. Shriberg, "Automatic Dialog Act Segmentation and Classification in Multi-party Meetings," In *Proc. Intl. Conf. on Acoustic, Speech and Signal Processing*, pages 1061–1064, 2005.
2. A. Clark and A. Popescu-Belis, "Multi-level Dialogue Act Tags," In *SIGdial Workshop on Discourse and Dialogue*, pages 163–170, 2004.
3. M. Cooke and D.P.W. Ellis, "The Auditory Organization of Speech and Other Sources in Listeners and Computational Models," *Speech Communication*, vol. 35, pages 141–177, 2001.
4. Ö. Çetin and A. Stolcke, *Language Modeling in the ICSI-SRI Spring 2005 Meeting Speech Recognition Evaluation System*, Technical Report TR-05-006, ICSI, 2005.
5. Ö. Çetin and E.E. Shriberg, "Speaker Overlaps and ASR Errors in Meetings: Effects Before, During, and After the Overlap," In *Proc. Intl. Conf. on Acoustic, Speech and Signal Processing*, 2006.
6. R. Dhillon, S. Bhagat, H. Carvey, and E. Shriberg, *Meeting Recorder Project: Dialog Act Labeling Guide*, Technical Report TR-04-002, ICSI, 2004.
7. G. Jefferson, "A Sketch of Some Orderly Aspects of Overlap in Natural Conversation," In G.H. Lerner (ed.) *Conversation Analysis*, pages 43–59, John Benjamins, 2004.
8. G. Ji and J. Bilmes, "Dialog Act Tagging Using Graphical Models," In *Proc. Intl. Conf. on Acoustics, Speech and Signal Process.*, pages 33–36, 2005.
9. A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI Meeting Corpus," In *Proc. Intl. Conf. on Acoustics, Speech and Signal Process.*, pages 364–367, 2003.
10. N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke "The Meeting Project at ICSI," In *Proc. Human Language Technologies Conf.*, pages 1–7, 2001.
11. NIST Speech Evaluations, `http://www.nist.gov/speech/tests/index.htm`.

12. T. Pfau, D. Ellis, and A. Stolcke, "Multispeaker Speech Activity Detection for the ICSI Meeting Recorder," *Proc. Automatic Speech Recognition and Understanding Workshop*, pages 107–110, 2001.
13. H. Sacks, E. Schegloff, and G. Jefferson, "A Simplest Semantics for the Organization of the Turn-taking in Conversation," *Language*, vol. 50, pages 696–735, 1974.
14. E. Schegloff, "Recycled Turn Beginnings: A precise repair mechanism in conversation's turn-taking organisation," In G. Button and J.R.E. Lee (ed.'s) *Talk and Social Organisation*, pages 70–85, Clevadon, 1987.
15. E. Schegloff, "Overlapping Talk and the Organization of Turn-Taking for Conversation," *Language in Society*, vol. 29, pages 696–735, 2000.
16. R.T. Schultz, A. Waibel, M. Bett, F. Metze, Y. Pan, K. Ries, T. Schaaf, H. Soltau, M. Westphal, H. Yu, and K. Zechner, "The ISL Meeting Room System," In *Proc. Workshop on Hands-Free Speech Communication*, 2001.
17. E. Shriberg, A. Stolcke, and D. Baron, "Observations on Overlap: Findings and implications for automatic processing of multi-party conversation," In *Proc. European Conf. on Speech Communication and Technology*, pages 1359–1362, 2001.
18. E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey, "The ICSI Meeting Recorder Dialog Act (MRDA) Corpus," In *Proc. 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100, 2004.
19. A. Stolcke, X. Anguera, K. Boakye, Ö. Çetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng, "Further Progress in Meeting Recognition: The ICSI-SRI Spring 2005 Speech-to-Text Evaluation System," In *Proc. NIST RT-05 Meeting Recognition Workshop*, 2005.
20. B. Wrede, S. Bhagat, R. Dhillon, and E. Shriberg. *Meeting Recorder Project: Hot Spot Labeling Guide*, Technical Repor TR-05-004, ICSI, 2005.
21. S. Wrigley, G. Brown, V. Wan, and S. Renals, "Speech and Crosstalk Detection in Multi-channel Audio," *IEEE Trans. on Speech and Audio Processing*, vol. 13, pages 84–91, 2005.
22. M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke, "A* based Joint Segmentation and Classification of Dialog Acts in Multiparty Meetings," In *Proc. Automatic Speech Recognition and Understanding Workshop*, pages 215–219, 2001.

A Speaker Localization System for Lecture Room Environment

Mikko Parviainen, Tuomo Pirinen, and Pasi Pertilä

Institute of Signal Processing, Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland
{mikko.p.parviainen,tuomo.pirinen,pasi.pertila}@tut.fi

Abstract. This paper presents a speaker localization system, which is an entry to Rich Transcription 2005 Spring Meeting Recognition Evaluation. The system is developed in the Institute of Signal Processing at Tampere University of Technology (TUT). The paper describes the framework of the evaluation and the proposed localization system. This paper is an extension to [1] giving the actual performance values of the system.

The localization system is based on spatially separate sensor stations. The sensor stations estimate Direction of Arrival (DOA) of acoustic wavefronts. Each sensor station produces a three dimensional DOA vector. The estimated DOA vectors at each time instant are combined to calculate the location of the sound sound source.

The performance of the system was determined using a set of predefined metrics. Using multiple metrics enables one to evaluate the performance of the localization system from different viewpoints. The overall performance is characterized by RMS error between estimates and reference positions. The results show that the performance of the proposed system is consistent and accuracy is satisfactory for meeting room scenario. However, several improvements can be seen.

1 Introduction

Localization of a sound source can be defined as determining its distance and direction with respect to a reference location and direction. An example of an advanced system is the human hearing. Humans with normal hearing perform sound source localization without an extra effort or decision. In fact, sound source localization is an integral part of our navigation and positioning system. Human hearing has other sophisticated capabilities such as separation of sound source and being able to understand another person's speech in noisy environments.

Localization has been a popular research topic over the years. Desire to obtaind knowledge of the human hearing system as such and foreseeable applications benefiting from location information have kept the research on this area active. For instance, hands-free devices may use location information to extract speech from background noise [2]. In a meeting room scenario, there are a variety of applications in which a sound source localization system may be useful.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 225–235, 2006.

Automatic translation to another language, retrieval of specific topics, and summarization of meetings in a human-readable form are a few desirable properties for a so called smart meeting room concept.

Despite the advances in research of human hearing and already existing applications, there are still shortcomings in the performance of systems trying to conduct the tasks in which humans are very good at. Human sound source localization system is a very complex process utilizing both acoustic and visual information. Additionally, learning and adaptation are involved while organizing auditory scene [3]. It is not know exactly how this information is utilized and combined.

The purpose of Rich Transcription Evaluation Series is to advance the technologies in the areas discussed above. Development in the areas within the framework of the evaluation may enable the deployment various applications including the ones discussed above. The focus is in human-to-human speech scenarios: broadcast news, conversational telephone speech and meeting room speech. "Black box" evaluations enable analysis of different systems to find their performance with a limited amount of prior information from the environment.

The paper is organized as follows. Section 2 presents briefly the Source Localization SLOC task. In Section 3, the speaker localization system is described. The real data results are presented and discussed in Section 4 and Section 5. Concluding remarks on the proposed sound source localization system are drawn in Section 6.

2 The Source Localization Task in 2005 Rich Transcription Evaluation

2005 Rich Transcription Evaluation focused on meeting domain and analysis on core speech technologies. Meeting domain offers a good environment to analyze and develop technologies for understanding human interaction. In this scenario, speech can be characterized as interactive and spontaneous. This means that there are several non-continuous sound sources in a relatively small enclosure. Effects of reverberation, ambient noise and sound source movement can be studied.

To address these issues, a Speech-to-Text (STT) transcription task and diarization task were established. Diarization is divided into three parts: (A) identifying a participant who is speaking and at which time, (B) Speech Activity Detection (SAD), and (C) Source Localization (SLOC). This paper describes briefly the SLOC task and the localization system developed for this task by Tampere University of Technology (TUT).

The task was arranged by a third party that provided objective performance evaluation of the localization systems. If data acquisition and the algorithm development were performed in a research group, it may happen that the data acquisition is conducted to favor a system. This can often happen even unconsciously. The arrangments of the evaluation enables a more truthful performance analysis. In this section SLOC task is briefly described to provide an overview of

the evaluation for the reader. The description here is based on the specifications of the evaluation.

2.1 The Objective

The objective set for the localization system is to estimate a three dimensional position of a single sound source. The sound source is a person, who is giving a lecture. To be more precise, this is a multi-source scenario, since the data consists of meeting sessions in a room with a video projector and other noise sources present. In this evaluation, a localization system is required to estimate the position of a single sound source. Other simultaneously active sound sources are regarded as noise sources and the system may disregard them and their positions. Additionally, the SLOC task is specified so that a localization system should produce an estimate only if the lecturer is speaking, other estimates are considered as false alarms.

2.2 Specification of Localization Task

The SLOC task is divided into two subtasks. The first is a *accurate localization task*, corresponding to situations when there is accurate reference information (true position) available. The second is a *rough localization task*, in which localization error can not be calculated, but it is likely that systems detect an acoustic event and thus are able to estimate its position. The reference data labeling controls which subtask is in question at a certain time instant.

2.3 Evaluation Criteria

Common evaluation criteria were established to enable "Black box" evaluation and to compare the localization systems developed in each institute participating to 2005 Rich Transcription Evaluation. A short review of the criteria and the metrics are presented here. A more detailed description can be found in [4].

Metrics. A brief introduction of metrics used in the evaluation is presented here. More details on metrics can be found in [5], [6]. The measured metrics were:

- $\mathbf{P_{cor}}$ — Localization rate (see description below)
- RMSE fine — RMS for fine error estimates
- RMSE fine + gross — RMS for fine and gross error estimates
- Bias fine — Bias for fine error estimates
- Bias fine + gross — Bias for fine and gross error estimates
- Deletion rate — Describes the amount of acoustic events left undetected
- False Alarm rate — Describes the amount of incorrect detections

RMS of Euclidean distance between the estimated coordinates and the reference coordinates is the most important metric. There are two types of errors. A *Fine Error* deviates 500 mm at a maximum for the accurate localization subtask

(RMSE fine). Otherwise it is a *Gross Error* (RMSE fine + gross). For the rough localization subtask the threshold between the error types is 1000 mm.

Additionally, the localization rate, $\mathbf{P_{cor}}$, is of interest. It is defined as

$$\mathbf{P_{cor}} = \mathsf{N_{FE}}/\mathsf{N_T}.$$

$\mathsf{N_{FE}}$ is the number of estimates classified within the fine error threshold and $\mathsf{N_T}$ is the total number of estimates produced by the localization system. The metrics are calculated over each meeting session. The total performance is estimated as average over the sessions.

2.4 Measurements

In this section the recording environment, content and equipment are described briefly. The data are comprised of meeting sessions that took place in a room equipped with microphones and other recording hardware. The recording sessions were organized by the CHIL project at The University of Karlsruhe. The data were distributed to the sites participating to the 2005 SLOC task.

The meetings were held in a 5.9 m × 7.1 m × 3.0 m room. The room is characterized as an ordinary room used for meetings and studying. Including the measurement equipment there is a table and a few seats for audience [7].

In each meeting there is a person giving a lecture and five persons as audience. The meetings were characterized as technical sessions, in which a person was presenting a topic. The lecturer was in front of the audience and was allowed to move around a small area.

Acquired data were divided into two sets: training data and evaluation data. Training data set enables each participant to test and tune their system with this type of data. The data are comprised of 13 meeting sessions. The duration of each meeting is approximately 15 minutes. In addition to the recorded signals, reference data were provided for each session. Training data were used to select the best processing methods and parameters for this kind of content.

Evaluation data set is the official data set to determine the performance of the systems. The data are comprised of 28 meeting excerpts. The duration varied from 69 to 806 seconds. The content of data can be divided into two types. In lecturer-only sessions, the lecturer was the dominating sound source. In questions-and-answers sessions, either the lecturer or a person from the audience was speaking. The lecturer-only sessions (a total of 13) were used to in the official performance analysis.

In 2005 Rich Transcription Evaluation data consist of video and audio data. The localization system developed by TUT uses only audio data.

There are three types of microphone setups in the recording sessions. The TUT localization systems uses the four microphone sensor stations mounted on each wall in the room. The purpose of the other setups is to acquire data for the other tasks of the 2005 Rich Transcription Evaluation [4]. Each sensor station contains four microphones. The shape of each station resembles an upside-down "T". Sensitivity pattern of the microphones is omnidirectional. The geometry of the sensor stations is illustrated in Fig. 1.

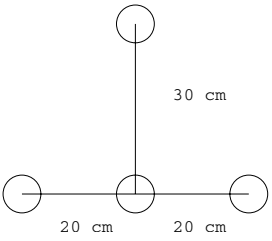


Fig. 1. The microphone geometry of the sensor stations used in the localization task. The stations are mounted at a 2.3 m height, one on each wall of the meeting room. The microphone sensitivity patterns are omnidirectional.

The sampling rate of 44.1 kHz and 24 bit resolution per sample were used. More detailed information on the room, sensor station placement, and the other microphone setups can be found in [7].

3 Localization System Overview

In 2005 Rich Transcription Evaluation, the localization task can be done using audio and video data. The system described here is an audio-only localization system. The block diagram of the localization system is illustrated in Fig. 2. The system consists of four stages. Time delays are estimated from microphone pairs, three dimensional direction of arrival vector estimates are calculated from using time delays and microphone geometry, and finally DOA vector estimates are filtered for possible outliers. DOA estimates from each station are combined in the actual localization subsystem to estimate the position of the sound source. Next, the techniques applied in each stage presented above are discussed in more detail.

3.1 TD Estimation

TDE subsystem computes propagation time of wavefront traveling between two microphones. In fact, the TDE subsystem performs the computation for all microphone pairs in a single sensor station. Each sensor station contains four microphones.

TDE is based on the Generalized Cross-Correlation (GCC) method [8]. Additionally, PHAT weighting is used. The TDE subsystem estimates weighted cross-correlation function from two microphone channels. The time lag at which a maximum of cross-correlation occurs is selected as time delay estimate. The weighted form of cross-correlation function is used to enhance the TDE process.

The TDE analysis window length was 32768 samples with 50 % overlap. A sensor station performs the analysis independently producing six time delay estimates. Thus, for each time instant, there are four sets of time delay estimates.

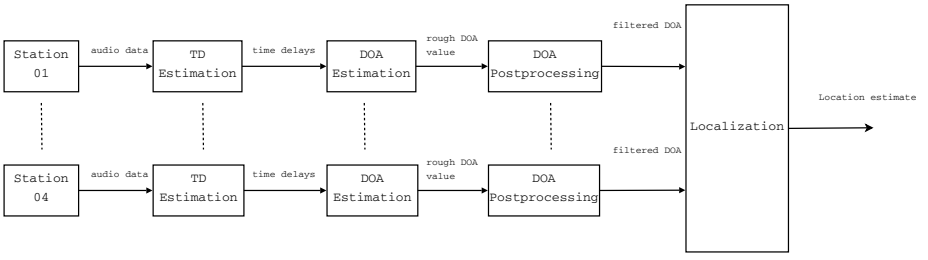


Fig. 2. Localization system receives its input from spatially separated four-microphone sensor stations. Each array estimates DOA from the wavefronts propagating towards the arrays. Four similar sensor stations were used to localize the active speaker [4].

3.2 DOA Estimation

The DOA subsystem is based on time delays estimated by TDE subsystem. In the DOA estimation a planar wave propagation model is assumed (far field case). There is no accurate definition for far field, but the assumption is plausible whenever the distances between a sound source and a sensor station is large compared to dimensions of the station (distance between microphones). Far field assumption enables the use of a low complexity DOA estimation technique. The localization system described here utilizes the method presented in [9]. The technique makes it possible to estimate the propagation vector ([10]) of the sound wave directly using only the estimated time delays and the sensor station geometry. The speed of sound varies depending on environment which is an error source in many DOA systems. On the other hand in a meeting room environment the effect of variation to location estimates is insignificant for most purposes. However, the method used in this localization system is independent of speed of sound information.

In this system, two time delays of six are used. The time delays to be used to estimate DOA is determined by their relative reliability. The reliability of TDE estimates was determined by a confidence scoring method proposed in [11], which is based on testing the linear dependence of the time delay estimates. DOA estimation and confidence scoring is conducted at each array.

3.3 DOA Postprocessing

Background noise may result in erroneous time delay estimation. For instance, TDE algorithm may select a local maximum, which is not resulting from the sound source of interest. Thus, the obtained DOA estimates can be classified as an outlier. To avoid these outliers, these propagation vectors are filtered out after the actual DOA estimation. For 2005 Rich Transcription Evaluation data a median filter was used. The length of the filter was nine frames, which was selected by analysis of the training data results. The filtering is conducted component-wise for the propagation vectors.

3.4 Localization

The localization subsystem is based on a bearings-only localization scheme. Bearings-only systems [12],[13],[14] estimate location of a sound source by combining DOA measurements from spatially separated locations. The localization system is independent of the preceeding processing stages (see Fig. 2). The only requirement is that DOA information is provided at the same rate from individual sensor stations.

If all stations could perform DOA estimation without errors, all DOA vectors would be associated to the same sound source in a single sound source scenario. Thus, lines drawn parallel to DOA vector estimates and passing through the position of the arrays would intersect at the true position of a sound source. However, estimated DOA information deviates from the actual DOA, it is only possible to estimate the location by minimizing a distance-based criterion. The location estimate is set to the point that minimizes the Euclidean distance. In this work, the distance is minimized in least squares sense. A closed-form solution is used [13].

3.5 Overall Execution Time

The evaluation data set contains roughly 2.5 hours of data. Taking into account all microphones in all sensor stations, a total of 40 hours data to be computed. The data set was run on a 2.8 GHz Pentium 4 processor with 2.1 GB of RAM. The processing time to compute the results for the evaluation data set was 10.8 hours, or 4.3 $\times$ real time. Most of the computation time is used by TD estimation stage.

The TUT localization system is implemented completely in Matlab environment. No external libraries or binaries were used in implementation.

4 Experiments

In this section the results obtained with the evaluation data and the training data sets are presented. In Rich Transcription Evaluation, the performance of the system is measured using the evaluation data set. The two data sets are discussed in Section 2.4. The results presented in this paper are different from the results that were presented in the actual evaluation. After the evaluation, it was observed that the evaluation data set was corrupted. This paper presents the results for the corrected evaluation data set.

The performance of the localization system is measured with a specific scoring tool designed for this kind of evaluations. The metrics and their motivation are explained in detail in [5]. Table 1 presents the results obtained from the evaluation and training data sets. The results for both data sets are averages for 13 meeting sessions.

Table 1 is obtained by first performing scoring each meeting session individually. The outputs of the scoring tool for each meeting session were fed to another tool to calculate the arithmetic mean for the metrics presented in the table.

Table 1. System performance for the evaluation and training data sets. The results are obtained for evaluation data set which is disjoint from the data used for training. The metrics are obtained as an average over 13 meeting sessions.

Metric	Training data set	Evaluation data set
$\mathbf{P_{cor}}$	0.43	0.62
RMSE fine + gross [mm]	812	749
RMSE fine [mm]	364	318
Bias fine [mm]	(186,23,112)	(185,61,12)
Bias fine + gross [mm]	(377,68,169)	(382,47,-54)
Deletion rate	0.00	0.00
False Alarm rate	1.00	0.98
N_T	28853	10854

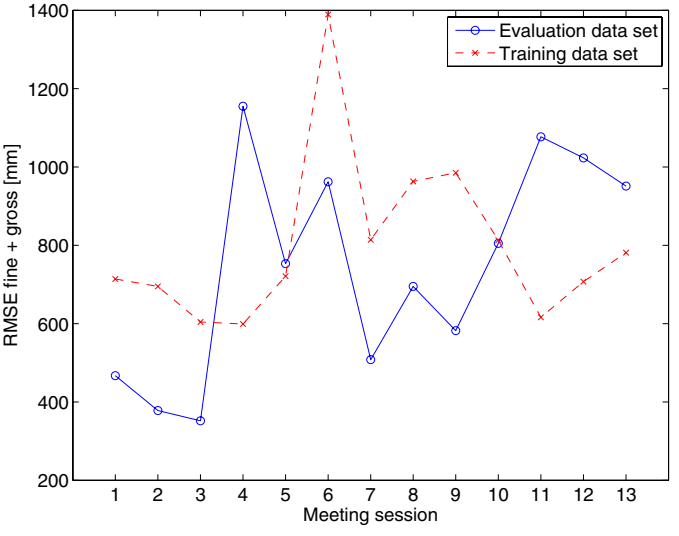


Fig. 3. RMSE fine + gross metric was evaluated from each meeting session. Note that the evaluation data set and the training data set are disjoint.

The performance of the localization system is best characterized RMSE metric, which is the root mean squared error in Euclidean distance between the reference location and the location estimated by the system. The estimated Bias is presented for each component (x, y, z). In this localization system there is no SAD subsystem. Thus, Deletion rate and False Alarm rate are not useful in analyzing this system.

A single recording contains a lot of data enabling the evaluation the performance of the localization system. Thus it useful to analyze the performance for each session separately. Fig. 3 presents RMSE fine + gross metric and Fig. 4 the $\mathbf{P_{cor}}$ metric.

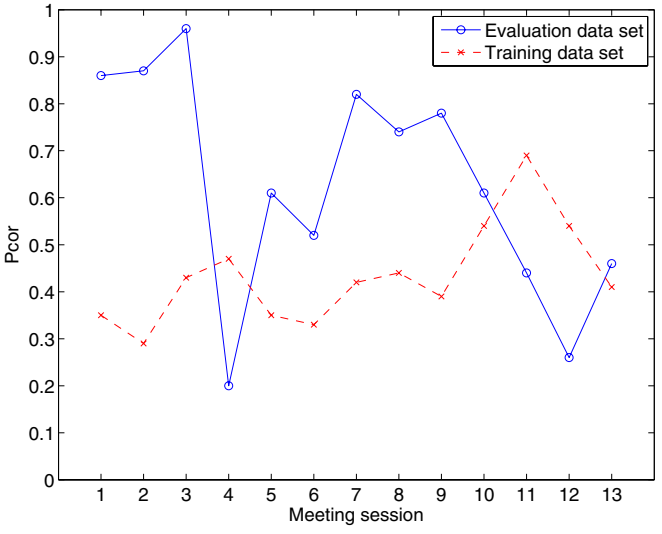


Fig. 4. $\mathbf{P_{cor}}$ metric for each meeting session. This metric describes, in a sense, robustness of a localization system. Values near one suggest better performance. Note that the evaluation data set and the training data set are disjoint.

5 Discussion

Reviewing the obtained results is a non-trivial task. However, while interpreting the results it should be noted that the localization system was tested in an environment from which there was a limited amount of information available. Usually, while algorithms are developed for such a task, hardware, placement of sensor stations, microphone configuration in a station and so forth are taken into account. In Rich Transcription Evaluation this can not be done since the measurements and algorithm development are separate processes. However, "Black-box" evaluation is fruitful since it forces towards developing a highly parameterized systems and limits the amount of optimization. System architectures developed in such a manner make them usable in various applications and environments.

In addition to the fixed measurement setup, common evaluation metrics were used. Multiple metrics, presented in Section 2, can be argued by thinking of a variety of applications. Accuracy of location estimates is used to measure sound source localization systems. However, accuracy may not be the most important in all scenarios. For instance, let us consider a teleconference meeting. In such a scenario real time tracking of a person's location may be more important criterion than a sheer accuracy of location estimates.

The performance of the localization system is best characterized by RMSE fine + gross and $\mathbf{P_{cor}}$ metrics. These metrics measure the performance in two senses. RMSE fine + gross measures the accuracy of a localization system by comparing the estimate to reference. $\mathbf{P_{cor}}$ can be viewed as a robustness

metric. RMSE fine + gross itself is a valid measure for accuracy. However, this is true only if the true location of a sound source is available. In 2005 Rich Transcription Evaluation, the annotations (including reference positions) were derived from video data.

RMSE fine + gross and $\mathbf{P_{cor}}$ metrics are illustrated in Fig. 3 and 4 for each session. It can be seen that the variation of the values is quite large between the recordings. There are multiple reasons to these variations. For instance, changes in generation process of the reference data, speaker, audience and noise sources affect the performance.

Table 1 presents the average values calculated over 13 recordings. It can be stated that the overall performance is satisfactory. Actually, the accuracy of the system may be even better than indicated by the table. There seems to be a bias in data. In general, a bias is removed before calculating actual performance values. However, the reason for the bias is unclear. It may result from the reference data, but there are other possibilities.

It has to be emphasized that the localization system presented in this paper does not have any SAD subsystem. The lack of this feature affects also the RMSE fine + gross and $\mathbf{P_{cor}}$ metrics. Using an appropriate method for SAD, it is likely that one would have obtained some gain in performance.

In summary, one can state that the framework for measuring performance is appropriate for the SLOC task. In a totally different scenario or goals, one has to review metrics to be used.

6 Conclusions

An acoustic speaker localization system was presented. The system is based on spatially separate sensor stations. Each sensor station is able to determine Direction of Arrival (DOA) of a sound source. DOA information is combined in a central unit to estimate the location of the sound source. The focus in this paper is in meeting domain but the system is scalable also to more distant sources.

The performance was measured by predefined metrics by a party outside the institute that developed the system. Thus optimization of the localization system was limited to calculation parameters.

The results are good, but still there is a lot of work to do. Since the content used in the experiments is mainly speech, using a Speech Activity Detection technique integrated into the localization system may increase the robustness against noise sources. It is likely that a performance gain would be achieved by using a tracking technique such as Kalman or particle filtering. For instance, tracking of DOA estimates may enhance robustness against outliers and thus prevents them from affecting location estimates.

References

1. Pirinen, T., Pertilä, P., Parviainen, M.: The TUT 2005 source localization system. In: "Rich Transcription 2005 Sring Meeting Recognition Evaluation", July 13, 2005, Royal College of Physicians, Edinburgh, UK. (2005)

2. Nakatani, T., Okuno, H.G.: Harmonic sound stream segregation using localization and its application to speech stream segregation. Speech Communication **27** (1999) 209 – 222
3. Bregman, A.S.: Auditory Scene Analysis. The MIT Press (1990)
4. National Institute of Standards and Technology: Spring 2005 (RT-05S) Rich Transcription Meeting Recognition Evaluation Plan. (2005) `http://www.nist.gov/speech/tests/rt/rt2005/spring/rt05s-meeting-eval-plan-V1.pdf`.
5. Omologo, M., Brutti, A., Svaizer, P.: Speaker localization and tracking – evaluation criteria. Technical report, National Institute of Standards and Technology (2005) `http://www.nist.gov/speech/tests/rt/rt2005/spring/sloc/CHIL-IRST_SpeakerLocEval-V5.0-2005-01-18.pdf1`.
6. Surcin, S., Stiefelhagen, R., McDonough, J.: D7.4 evaluation packages for the first CHIL evaluation campaign. Technical report, Computers in the Human Interaction Loop (CHIL) Consortium (2005)
7. Stiefelhagen, R.: CHIL evaluation data – overview of sensor setup and recordings. Technical report, Computers in the Human Interaction Loop (CHIL) Consortium (2004)
8. Knapp, C., Carter, G.C.: The generalized correlation method for estimation of time delay. IEEE Transactions on Acoustics, Speech, and Signal Processing **24**(4) (1976) 320–327
9. Yli-Hietanen, J., Kalliojärvi, K., Astola, J.: Low-complexity angle of arrival estimation of wideband signals using small arrays. In: Proceedings of the 8th IEEE Signal Processing Workshop on Statistical Signal and Array Signal Processing. (1996) 109–112
10. Haykin, S., Justice, J.: Array signal processing. Academic Press (1985)
11. Pirinen, T.: Normalized confidence factors for robust direction of arrival estimation. In: Proceedings of the 2005 IEEE International Symposium on Circuits and Systems (ISCAS). (2004)
12. Kaplan, L., Le, Q., Molnár, P.: Maximum likelihood methods for bearings-only target localization. In: Proceedings of the 2001 IEEE International Conference on Acoustics Speech, and Signal Processing (ICASSP '01). (2001) 3001 – 3004
13. Hawkes, M., Nehorai, A.: Wideband source localization using a distributed acoustic vector-sensor array. IEEE Transactions on Signal Processing **51**(6) (2003) 1479 – 1491
14. Pertilä, P., Parviainen, M., Korhonen, T., Visa, A.: A spatiotemporal approach to passive sound source localization. In: International Symposium on Communications and Information Technologies (ISCIT 2004). (2004)

Robust Speech Activity Detection in Interactive Smart-Room Environments

Dušan Macho, Climent Nadeu, and Andrey Temko

TALP Research Center, Universitat Politècnica de Catalunya, Campus Nord, Edifici D5
Jordi Girona 1-3, 08034 Barcelona, Spain
{dusan,climent,temko}@talp.upc.edu

Abstract. In perceptive interface technologies used in smart-room environments, the determination of speech activity is one of key objectives. Due to the presence of environmental noises and reverberation, a robust Speech Activity Detection (SAD) system is required. In a previous work, a SAD system, which used Linear Discriminant Analysis-extracted features and a Decision Tree classifier, was successfully contrasted with other previously reported techniques in a set of room environment tests with the SPEECON database. In this work, the same SAD system has been tested in even more realistic conditions involving meetings, and it has been modified to significantly improve its performance. Actually, we have trained the SAD system with a subset of SPEECON data, and without any further tuning we have used it to carry out tests with the meeting databases from the NIST RT05 evaluation. In order to improve the SAD performance, we consider two additional features which are measures of energy dynamics at low and high frequencies, respectively. Besides that, two alternative classifiers have been tested, which are based on Support Vector Machines and Gaussian Mixture Models, respectively. With the latter classifier, and using both the LDA features and the low-frequency energy dynamics feature, a large improvement in speech detection performance has been observed, e.g. the NIST error rate was reduced from 20.69% to 8.47% for the RT05 evaluation data. In addition, we report the results obtained with a slightly modified version of the SAD system in the NIST RT06 evaluation.

1 Introduction

Detecting the presence of speech is a key objective in speech-related technologies. In fact, Speech Activity Detection (SAD) usually allows an increase of recognition rate in automatic speech or speaker recognition, and it is also required in both speech/speaker recognition and speech coding to save computational resources (and batteries) in the devices where the processing of non-speech events is not needed. Also, as many speech enhancement techniques require a proper estimate of noise characteristics, the reliable detection of non-speech portions of signal is needed. On the other hand, SAD may boost the performance measures of other technologies belonging to audio scene analysis, like speaker localization or acoustic event detection. Last but not least in perceptive interface technologies, the determination of

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 236 – 247, 2006.

speech activity in a room environment may be used to infer the type of activity that takes place in the room, or at a specific position of the room, given the coordinates of the microphones whose signals show the presence of speech utterances.

Our work, inserted in the CHIL (Computers in the Human Interaction Loop) project framework, assumes a meeting room environment, where audio acquisition is done in an unobtrusive way by a network of far-field microphones. In such a challenging environment, a high robustness of the SAD algorithm against environmental noises and reverberation is extremely important. On the other hand, the working scenarios require online implementations that can operate in real time and only a given maximum latency is accepted. Consequently, segmentation algorithms that use the entire recorded file must be avoided.

In a previous work at our laboratory, we proposed a SAD algorithm [1] that assumed this kind of environment and working conditions. It was compared with other previously reported techniques using a subset of the SPEECON database [2]. The speech detection system was based on speech features that had already shown good robustness properties in automatic speech recognition: the Frequency-Filtered (FF) log spectral energies. The fact that these features are also used for speech recognition avoids the need to re-compute them for SAD when both tasks are being performed at the same time. The FF parameters were further processed by Linear Discriminant Analysis (LDA) to select only one feature per frame, and a Decision Tree (DT) classifier used a time sequence of these features to make the Speech/Non-Speech decision.

In this paper, further work is presented along that line. The already existing algorithm has been tested in more real conditions involving interactions of several persons in meetings and it has been modified to significantly improve its performance. We trained our SAD system with the previous subset of SPEECON and, without any additional tuning, we have used it to carry out tests with the meeting databases from the NIST Rich Transcription 2005 (RT05) evaluation. Both the usual NIST metrics and the ones used in CHIL for SAD have been used to compare performances. In order to improve the SAD results, we have considered two additional features which are measures of energy dynamics at low and high frequencies, respectively. Besides that, two alternative classifiers have been tested, which are based on Support Vector Machines (SVM) [7] and Gaussian Mixture Models (GMM) [9], respectively. The best SAD system was evaluated in the NIST Rich Transcription 2006 (RT06) evaluation.

The databases for training and testing are presented in Section 2. Section 3 describes the features and Section 4 is dedicated to the classifier training procedures. Experiments and results are presented in Section 5.

2 Databases

For the classifier training, we used a portion of the office environment recordings from the *Spanish* language SPEECON database [2]. In total, 90 minutes of signal recorded by a far-field omni-directional microphone placed 2-3 meters in front of the speaker was used. The training material was well balanced in terms of the two classes of interest; it contained 49% of Speech and 51% of Non-Speech. The

database sampling frequency was 16 kHz and the sample representation is 16 bits. Across all recordings, the audio signal uses about 50% of the available 16-bit dynamic range.

For the classifier test/development, we used the single distant microphone evaluation data from the NIST RT05 "conference room" meeting task [3]. It contains 10 extracts from 10 *English* language meetings recorded at 5 different sites. Each extract is about 12 minutes long. The proportion of Speech / Non-Speech is highly unbalanced, approximately 90% of all signal is Speech. The sampling frequency and sampling representation are the same as in the training data, 16 kHz and 16 bits, respectively. Some extracts, however, use only a small portion of the available dynamic range (less than 20%).

The RT06 test data set consists of two kinds of data, "confmtg" and "lectmtg". The confmtg data set is similar to the previously described RT05 data. The lectmtg data were collected from lectures and interactive seminars across the smart-rooms of different CHIL project partners.

The training and development/testing data are similar in a way that they are recorded in a closed environment using a far-field microphone, thus the recordings have a relatively low SNR due to the reverberation and the environmental noise. However, there are some differences that should be mentioned: different language (Spanish vs. English), different setup of the acquisition hardware, different Speech and Non-Speech proportion. Also, it is worth to mention that the main task, and thus the main attention, of the speaker in the training database was the recording itself, while in the test meeting/lecture database, the recording was secondary. As a consequence, the test/lecture database is more spontaneous, speakers speak not necessarily heading the microphone, and the data contain overlapped speech.

3 Features

We investigate two kinds of features. The first feature set, based on Linear Discriminant Analysis (LDA) [9] of parameters that model spectra, extracts the information about the spectral shape of the acoustic signal from a short interval (approx. 70 ms). The second feature set focuses more on the dynamics of the signal along the time observing low- and high-frequency spectral components along a bit longer time interval (approx. 130 ms).

3.1 LDA Measure

The LDA measure, *ldam*, is based on Frequency Filtering (FF) features – a speech representation originally designed for ASR that showed higher robustness in noisy ASR tests than the usual mel-frequency cepstrum (MFCC) features (see e.g. [4]). The robustness issue is very important due to the low SNR of the recordings in our task.

The FF feature extraction scheme used in this work consists in calculating a log filter-bank energy vector of 16 bands for each signal frame (with frame length/shift = 30/10ms) and then applying a FIR filter with impulse response $h(k)=\{1, 0, -1\}$ on this vector along the frequency axis. The obtained static FF feature vector is accompanied with a short-time dynamic representation in form of delta (50ms) and delta-delta

(70ms) features. In addition, the delta of the frame energy is also appended. The size of the FF representation (16+16+16+1=49) is reduced to a single scalar measure by applying LDA, a data-driven linear transformation designed to extract the principal components of the input data using a discriminative criterion. That single LDA measure, *ldam*, is computed by multiplying the FF feature vector and the LDA eigenvector corresponding to the largest LDA eigenvalue as calculated from the training set. More details on the LDA FF features can be found in [1], where it is also shown that the FF+LDA measure is more discriminative than the MFCC+LDA measure.

3.2 Low-Frequency and High-Frequency Energy Dynamics Feature

In addition to the LDA measures, we experimented with two sub-band energy based features, low-frequency and high-frequency energy dynamics feature (*lfed* and *hfed*, respectively). *lfed* is calculated as follows:

$$E_l(t) = \log\left(\sum_k S(k,t)\right) \text{ where } 13 \le k \le 38 \tag{1}$$

$$dE_l(t) = \frac{1}{60}\sum_{i=-4}^{4} i \cdot E_l(t+i) \tag{2}$$

$$lfed(t) = \frac{1}{5}\sum_{i=-2}^{2} abs(dE_l(t+i)) \tag{3}$$

where $S(k,t)$ is the k-th bin of the FFT-512 power spectrum at the frame index t. *lfed* involves approximately a frequency range from 400 Hz to 1200 Hz comprising most of the interval of high energy concentration of the voiced speech sounds (sampling frequency of 16 kHz is assumed). *hfed* is calculated in the same way but $144 \le k \le 208$, which correspond to the interval from 4500 Hz to 6500 Hz and this feature focuses on fricative sounds. The frequency intervals of both features are based on general knowledge and were not tuned to the application. A similar feature as *lfed* and *hfed* was proposed in [5] but that feature was calculated over the entire frequency range and it included spectral autocorrelation to emphasize the speech harmonic structure.

Both *lfed* and *hfed* represent the amount of energy changes in the corresponding frequency bands. Figure 1 shows the evolution of all three features *ldam*, *lfed* and *hfed* along the time in a SPEECON Spanish sentence “Bajo los gestos fieros, late una terrible desesperanza.”. Notice that the energy dynamics features represent very well the speech onsets in their corresponding bands, which might be beneficial complementary information for the LDA feature which represents onsets in a softer way.

Notice that in the final signal representation, the contextual information is involved in several ways. First, before applying the LDA transform, the current delta and

delta-delta feature involve an interval of 50 and 70 ms, respectively, in their calculation. Next, for the representation of the current frame, eight LDA measures are selected from a time window spanning the interval of 310 ms around the current frame. Finally, *lfed* and *hfed* involve a smoothed derivative calculation that in total uses an interval of 130 ms.

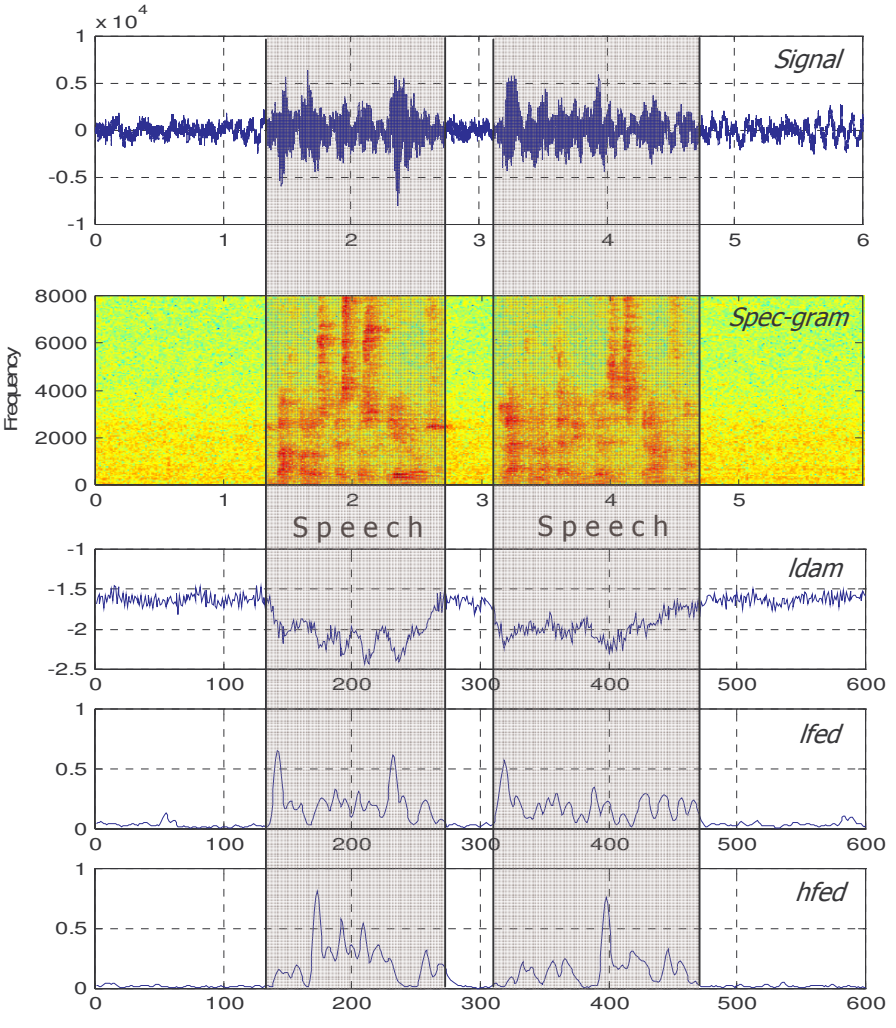


Fig. 1. From top down: audio signal, spectrogram, LDA measure feature, low-frequency and high-frequency energy dynamics features

We use the SAD system on-line in our smart-room, so we avoid using techniques that would cause an algorithmical delay larger than a given acceptable value (set to 160 ms in our case). In addition, the designed SAD feature extraction saves

computational resources since most of the calculation dedicated to the feature extraction is performed anyway for the ASR system due to the fact that SAD features are based on ASR features.

4 Classifier Training

In this section, we explain the training procedures used for the three classifiers used in this work. We use the Decision Tree (DT) as our baseline classifier and we contrast its performance with another discriminative classifier based on Support Vector Machine (SVM) [7] approach and a generative classifier based on Gaussian Mixture Model (GMM) [9].

4.1 Features

For each frame at a time index t, one LDA measure $ldam(t)$ is available. To include information from a time span larger than just 70ms into the representation, the eight most important LDA measures are selected from the interval $t-15 \le t \le t+15$. As a criterion for this selection, we used the entropy based information gain criterion used in the DT training algorithm (see [1] for more details on this selection process). LDA measures were concatenated to form the final representation vector

$ldam(t\text{-}15)\ ldam(t\text{-}10)\ ldam(t\text{-}6)\ ldam(t\text{-}3)\ ldam(t)\ ldam(t\text{+}3)\ ldam(t\text{+}6)\ ldam(t\text{+}10)$

Using these features, we defined the following four different feature sets (in parenthesis is the feature vector size):

A: Eight *ldam* features selected using the C4.5 DT training (8 features)
B: A + *lfed* (9 features)
C: A + *hfed* (9 features)
D: B + *hfed* (10 features)

The A feature set is considered a baseline in our tests (a six-feature version of this set was already used in [1]). The feature sets B, C, and D allow us to observe the contribution of the low and high frequency dynamics features when added individually to the feature set A, as well as when both of them are added to A.

The 90 minute training data, processed on frame-by-frame basis using a frame shift of 10 ms, results in over 500 thousand training examples with their corresponding Speech / Non-Speech labels. The Speech / Non-Speech labeling was performed by applying a forced Viterbi alignment on the training files using our speech recognition system.

4.2 Decision Tree (DT) Classifier

For DT training we used the C4.5 algorithm [6] which is an improvement of Quinlan's original ID3 DT training algorithm. During training, for each node of the decision tree, the best feature element from the feature vector is selected and the best

threshold is set for this element. Using the SPEECON training example set and setting the pruning confidence level to 25% resulted in decision trees with the following number of nodes, depending on the feature set: A 1031, B 1475, C 1571, and D 2117.

4.3 Support Vector Machine (SVM) Classifier

A set of 500 thousand of examples is an enormous number of feature vectors to be used for the usual SVM training approach and hardly makes such training process feasible in practice. Alternative methods should be used; we tested the so called cascade learning [8], however our implementation did not achieve a satisfactory performance. A better result was obtained by imposing a hard data reduction by randomly selecting 20 thousand examples where the two classes of interest are equally represented. The training data were firstly normalized anisotropicly to be in the range from –1 to 1, and the obtained normalizing template was then applied also to the testing data set. We used the Gaussian kernel with gamma parameter equal to 5.0 and the C parameter (controlling the training error) equal to 10.0. To train the system we use the publicly available SVMlight software package[1]. As a result, we obtained classifiers with the following numbers of support vectors, depending on the feature set: A 5170, B 4553, C 4829, and D 4447.

4.4 Gaussian Mixture Model (GMM) Classifier

We used the well known Expectation-Maximization (EM) [10] algorithm for Gaussian mixture model training with the K-means algorithm for the model parameter initialization. The number of mixtures was set to 32 for both Speech and Non-Speech classes and diagonal covariance matrices were used. 20 iterations of the EM algorithm were performed.

5 Experiments

5.1 Metrics

We present results using several metrics; as a primary metric we use the one defined for the SAD task in the NIST Rich Transcription evaluation. It is defined as the ratio of the duration of incorrect decisions to the duration of all speech segments in reference. We denote this metric as NIST in our results.

Notice that the NIST metric depends strongly on the prior distribution of Speech and Non-Speech in the test database. For example, a system that achieves a 5% error rate at Speech portions and a 5% error rate at Non-Speech portions, would result in very different NIST error rates for test databases with different proportion of Speech and Non-Speech segments: in the case of 90-to-10% Speech-to-Non-Speech ratio the NIST error rate is 5.6%, while in the case of 50-to-50% ratio it is 10%. Due to this fact we report three additional metrics[2]: Mismatch Rate (MR), Speech Detection Error Rate (SDER), and Non-Speech Detection Error Rate (NDER) defined as:

[1] SVMlight: http://svmlight.joachims.org/

[2] These metrics were used within internal CHIL project evaluation.

- MR = Duration of Incorrect Decisions / Duration of All Utterances
- SDER = Duration of Incorrect Decisions at Speech Segments / Duration of Speech Segments
- NDER = Duration of Incorrect Decisions at Non-Speech Segments / Duration of Non-Speech Segments

5.2 Results on SPEECON Data

First, we tested the SAD systems on the SPEECON training data. Table 1 shows results when using the four feature sets A, B, C, D for each of the three classifiers. The primary error rates were calculated using the NIST metric and the other metrics are mentioned below it so that the error for Speech and Non-Speech class can be observed separately.

Adding the *lfed* feature (column B) as well as adding the *hfed* feature (column C) improves the speech detection performance in comparison to using the LDA features only (column A). Notice that both SDER and NDER are reduced and, thus, the total improvement is not achieved by having a cost on one of the two classes. This improvement and the fact that the error is reduced in all tested classifiers make *lfed* and *hfed* in combination with *ldam* features an interesting feature set for SAD. Adding *lfed* shows larger benefit than adding *hfed*, which can be accounted for the larger speech information content in the low acoustic frequencies than in the high frequencies. The best result for each classifier is achieved when adding both features simultaneously (column D).

Table 1. Error rates obtained for the SPEECON train data

	NIST MR / SDER / NDER			
Feat set	A	B	C	D
DT	20.79 9.21 / 11.51 / 8.94	17.56 7.80 / 10.27 / 7.02	18.51 8.16 / 10.15 / 8.04	16.10 7.17 / 9.37 / 6.71
SVM	21.63 10.54 / 11.25 / 9.86	18.29 8.91 / 9.95 / 7.93	19.43 9.47 / 9.99 / 8.96	17.15 8.36 / 9.08 / 7.67
GMM	24.11 11.75 / 12.90 / 10.65	20.49 9.98 / 11.65 / 8.40	21.52 10.49 / 12.15 / 8.91	19.28 9.39 / 10.87 / 7.99

The DT discriminative classifier shows the best performance for all tested feature sets. The DT classifier is the one most tuned to the used features – recall that the eight LDA features were selected using the DT training algorithm and were used for the other two classifiers without any modification. Also we spent much more time working with DT in the context of this training database than with the other two classifiers, so from there may come some degree of tuning of DT to the SPEECON

database. Nevertheless, the performance of SVM is very close to the performance of DT. On the other hand, the GMM generative classifier with 32 mixtures performs notably worse than DT. A GMM system using 64 mixtures did not bring any additional gain.

5.3 Results on RT05 Data

For the RT05 test database, a post-processing was applied to each SAD output consisting of marking the non-speech intervals shorter than 0.3 seconds as speech. This non-speech gap smoothing was used to mimic the same post-processing that was applied to the original human labels used as the reference.

Notice also that the test database contains much more Speech intervals than Non-Speech intervals (approx. 90% Speech). The NIST metric is inversely proportional to the amount of Speech in the reference labels. Thus, assuming the same amount of incorrect decisions in both a testing data with 50% Speech content and a testing data with 90% Speech content, a lower NIST error will be reported for the later testing data. This is the reason why some error rates reported for the unbalanced test database (RT05) may be lower that those reported for the more balanced train database (SPEECON).

Table 2 shows the results we obtained on the RT05 test database. Neither features nor classifiers were tuned to these test data. Most of the observations about the *lfed* and *hfed* features from the previous experiments with the training data hold also in this case, which is quite encouraging. An exception is the feature set D in the GMM classifier, where adding *hfed* does not improve the performance of the feature set B for the same classifier. In general, significant improvements can be seen when adding the *lfed* and *hfed* features to the LDA vector; for example in the case of SVM, the usage of both features reduces the original error by 52%. Among the classifiers, GMM achieves substantially lower error rates than the two discriminative classifiers (note that GMM was the worst performing on the training data). It seems in our case that the GMM classifier generalizes better the knowledge from the training data than

Table 2. Error rates obtained for the RT05 test data

	NIST MR / SDER / NDER			
Feat set	A	B	C	D
DT	20.69 18.77 / 18.21 / 24.32	12.37 11.20 / 9.24 / 30.51	14.76 13.37 / 11.65 / 30.27	11.54 10.43 / 8.10 / 33.42
SVM	23.88 21.71 / 21.46 / 24.22	14.70 13.36 / 11.87 / 28.19	15.69 14.26 / 12.51 / 31.76	11.45 10.41 / 7.99 / 34.56
GMM	12.25 11.13 / 8.86 / 33.81	8.47 7.69 / 4.61 / 38.42	10.02 9.11 / 5.24 / 47.70	8.66 7.88 / 3.75 / 49.00

the two other classifiers. The performance of DT and SVM is very similar, especially in the D feature set case. The best overall performance, NIST error of 8.47%, was obtained using the GMM classifier with the feature set B. It represents a 59% error rate reduction with respect to the performance of the baseline system consisting of the DT classifier and the feature set A (20.69%). Notice the NDER scores are high in comparison to the SDER scores in these tests. This is caused mostly by the combined effect of the Non-Speech gap post-processing and the low amount of the Non-Speech testing material which was mentioned above.

5.4 Results on RT06 Data

In this subsection we present the results achieved by our SAD system in the RT06 evaluation campaign. Based on our previous experiments, we selected the GMM classifier. As for features, we used the feature set D augmented by a cross-frequency energy dynamic feature, *xfed*, which is obtained as a combination of *lfed* and *hfed* and it is calculated as follows:

$$xfed(t) = \frac{\sqrt{hfed(t-9)\cdot lfed(t+9)} + \sqrt{hfed(t+9)\cdot lfed(t-9)}}{2} \tag{4}$$

This feature reaches high values when both the *hfed* before and *lfed* after (or *hfed* after and *lfed* before) the current frame have high values. It attempts to follow the energy flow between low and high frequencies typical for speech. The 9 frame distance was set empirically (it would correspond to 1/0.18 = 5.6 Hz energy flow rate) and it is limited by the maximum allowed algorithmical delay of the SAD system.

For the confmtg task, both SPEECON and RT05 databases were used to train the SAD system. For the lectmtg task, also a small amount of CHIL data was added into the training of the final system.

The RT06 evaluation campaign has several evaluation subtasks, depending on the used set of microphones. We participated in two subtasks, single distant microphone denoted as *sdm* and multiple distant microphones, *mdm*. The *sdm* subtask involved the centrally located omni-directional table microphone, or the best microphone selected after listening to the recordings. The *mdm* subtask involved at least 3 omni-directional table microphones, including the one selected for the *sdm* subtask.

In the tests with the RT06 data we applied a slightly different post-processing on top of the classifier output than we did in the case of the RT05 data tests. First, we performed a majority voting, where the central frame of an 11 frame interval was marked as Speech if 6 or more frames of this interval were classified as Speech; otherwise it was marked as Non-Speech. Second, to each Speech segment, 0.2 s of Speech was added at the beginning and the end of segment.

In the case of *mdm*, we applied a separate SAD to each individual channel, without post-processing. Outputs of all SAD systems were merged by a majority voting performed for each frame favoring the Speech label in the case of a tie. Then the same post-processing as in the *sdm* case was applied on the output of merging.

Table 3 shows error rates obtained by our SAD system in the RT06 evaluation tasks. Very low NIST error rates were obtained and our SAD system ranked among

Table 3. Error rates obtained for the RT06 evaluation tasks

Feat set D + *xfed*	NIST MR / SDER / NDER	
	confmtg	lectmtg
sdm	5.45 5.1 / 3.1 / 41.4	7.10 6.2 / 0.4 / 48.1
mdm	5.63 5.3 / 3.5 / 38.7	5.30 4.6 / 0.7 / 33.3

the best systems; however, the Non-Speech detection error rates are high. To reduce this error rate will be the objective of our future work. We could benefit from the multiple microphones in the case of lectmtg task, however there is no significant change in the case of confmtg task.

6 Conclusion

The presented work is oriented towards a robust Speech Activity Detection (SAD) in smart-room environments. The baseline SAD system used features obtained by applying Linear Discriminative Analysis (LDA) on a parameter set modeling the shape of the signal spectrum; Decision Tree was used to perform the Speech/Non-Speech classification on a frame-by-frame basis. Both the LDA transform and the Decision Tree classifier were trained with a portion of the Spanish SPEECON database. The SAD system was evaluated on an interactive meeting database from the NIST RT05 evaluation campaign, as well as on the RT06 evaluation corpus.

In this work we improved significantly the performance of the baseline speech detector. We tested additional features that measure the signal energy dynamics at low and high frequencies. Also, two other classifiers were evaluated for the SAD task: a discriminative Support Vector Machine classifier and a generative Gaussian Mixture Model (GMM) classifier. We observed that appending the high and low frequency energy dynamics features to the LDA features improved the performance for both training and testing data across all the classifiers with a higher benefit from the low-frequency feature. The highest NIST error rate reduction was achieved by using the GMM classifier and the LDA features with the low-frequency energy dynamics; the error of the baseline SAD was reduced from 20.69% to 8.47% for the RT05 evaluation data. Very competitive results were also obtained with a slightly modified system in the RT06 SAD evaluation tasks.

Acknowledgements

The authors wish to thank Jaume Padrell for encouraging discussions on the topic. This work has been partially sponsored by the EC-funded project CHIL (IST-2002-506909) and the Spanish Government-funded project ACESCA (TIN2005-08852).

References

1. Padrell J., Macho D., Nadeu C., "Robust Speech Activity Detection Using LDA Applied to FF Parameters", Proc. ICASSP'05, Philadelphia, PA, USA, March 2005.
2. Iskra D. J. et al., "SPEECON - Speech Databases for Consumer Devices: Database Specification and Validation", Proc. LREC, 2002.
3. Fiscus J.G., Radde N., Garofolo J.S., Le A., Ajot J., Laprun C., "The Rich Transcription 2005 Spring Meeting Recognition Evaluation", Lecture Notes in Computer Science (LNCS), vol. 3869, pp.369-389, Springer, February 2006.
4. Nadeu, C., Macho, D., and Hernando, J., "Frequency and Time Filtering of Filter-Bank Energies for Robust HMM Speech Recognition", Speech Communication, Vol. 34, pp. 93-114, 2001.
5. Ouzounov A., "Robust Feature for Speech Detection", Cybernetics and Information Technologies, vol.4, No.2, pp.3-14, 2004 (http://www.iit.bas.bg/staff_en/ SpeechDetectionFeature.pdf).
6. Quinlan, J. R., "C4.5: Programs for Machine Learning", Morgan Kaufmann, 1992.
7. B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, Cambridge, MA, 2002.
8. Graf H.P., Cosatto E., Bottou L., Durdanovic I., Vapnik V., "Parallel Support Vector Machines: The Cascade SVM", Proc. Eighteenth Annual Conference on Neural Information Processing Systems, 2004.
9. Duda R., Hart P., Stork D., Pattern Classification, 2nd Edition, Wiley-Interscience, 2000.
10. Rabiner L., Juang B.H., Fundamentals of Speech Recognition, Prentice Hall, 1993.

Automatic Cluster Complexity and Quantity Selection: Towards Robust Speaker Diarization

Xavier Anguera[1,2], Chuck Wooters[1], and Javier Hernando[2]

[1] International Computer Science Institute, Berkeley CA 94704, USA
[2] Technical University of Catalonia, Barcelona, Spain
{xanguera,wooters}@icsi.berkeley.edu

Abstract. The goal of speaker diarization is to determine where each participant speaks in a recording. One of the most commonly used technique is agglomerative clustering, where some number of initial models are grouped into the number of present speakers. The choice of complexity, topology, and the number of initial models is vital to the final outcome of the clustering algorithm. In prior systems, these parameters were directly assigned based on development data, and were the same for all recordings. In this paper we present three techniques to select the parameters individually for each case, obtaining a system that is more robust to changes in the data. Although the choice of these values depends on tunable parameters, they are less sensitive to changes in the acoustic data and to how the algorithm distributes data among the different clusters. We show that by using the three techniques, we achieve an improvement up to 8% relative in the development set and 19% relative in the test set over prior systems.

1 Introduction

The goal of speaker diarization is to segment an audio recording into speaker-homogeneous regions [1]. Typically, this segmentation must be performed with little knowledge of the characteristics of the audio or of the participants in the recording. For example, we may know the source and date of the audio recording (e.g. CNN Nightly News or a NIST meeting), but we typically do not know how many speakers occur in the recording, how many speakers are male vs. female, whether there are commercials, music, or other noises, etc.

Typically, most speaker diarization systems use algorithms that are governed by tunable low-level parameters that are adjusted using development data of the same sort as the testing data. This is the case, for example, for the acoustic models used, the penalty factor used on the Bayesian Information Criterion (BIC) to compare models [2], and some initial parameter values such as the number of initial speaker clusters, the number of Gaussian mixtures per model at each state of the process, and the average speaker turn length. Such systems perform poorly when conditions change between the train and test sets; also selecting a constant value for all the recordings in a set lead to the omission of some particularities of each recording, resulting in a suboptimal result.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 248–256, 2006.

In this paper, we present three algorithms that help determine important parameters in the clusters modeling, namely the number of Gaussian mixtures per model at each step of the processing, the number of initial models in the system, and the topology of each acoustic model. In order to determine the number of Gaussian mixtures and the number of initial clusters, the algorithms base their selection on information on each particular recording rather than defining a pre-fixed value for all recordings of a certain type. In order to do this, we define a parameter that we call the Cluster Complexity Ratio (CCR), which defines a ratio between the data being modeled and the mixtures needed to represent it. The CCR ratio is defined using development data, and it is used to define recording-specific values for the above mentioned parameters.

The third novelty presented in this paper is the elimination of the dependency of the acoustic models on the average speaker turn length. This is achieved by modifying the acoustic modeling topology by changing the probabilities of self-loop and transition in the last state. By doing so, we can apply a minimum duration for a speaker turn while not influencing the final duration. While setting a minimum duration for speaker turns is advantageous for the processing of the recordings and can be set to be independent of the kind of recording we encounter, the average speaker turn duration is quite variable between individual recordings and recording types. It is therefore interesting to let the acoustic data define when the speaker turn finishes once it achieves a minimum length.

In section 2, we present the speaker diarization algorithm with the proposed algorithms. In sections 3 through 5, we present the algorithms in detail. Then the experiments are presented, and finally conclusions are drawn from them.

2 Agglomerative Speaker Diarization System

As explained in [3] and [4], the speaker clustering system is based on an agglomerative clustering technique. It initially splits the data into K clusters (where K must be greater than the number of speakers and is chosen by the presented algorithm), and then iteratively merges the clusters (according to a merge metric based on ΔBIC) until a stopping criterion is met. Our clustering algorithm models the acoustic data using an ergodic hidden Markov model (HMM), where the initial number of states is equal to the initial number of clusters (K). Upon completion of the algorithm's execution, each remaining state is taken to represent a different speaker. Each state in the HMM model contains a set of MD sub-states, imposing a minimum duration on the model (we use $MD = 3$ seconds). Within the state, each one of the sub-states shares a probability density function (PDF) modeled via a Gaussian mixture model (GMM). A modification to this architecture presented in this paper avoids any maximum time duration constraints on the speaker turns, as further explained in section 5.

The following items show step by step the clustering algorithm used in the meetings domain, where we include the novel processing presented in this paper (explanation on previous systems can be found in [5] and [3]):

1. Run speech/non-speech detection on input data.
2. Extract acoustic features from the data and remove non-speech frames.
3. **(new)** Estimate the number of initial clusters K and set their initial model complexity (number of Gaussian mixtures per model).
4. Create models for the K initial clusters via linear initialization.
5. Perform several iterations of segmentation and training to stabilize the data among the different models.
6. **(new)** Adjust the complexity of each resulting model according to the data assigned to them and retrain all models.
7. Perform iterative merging using the following steps:
   (a) Run a Viterbi decode to resegment the data.
   (b) **(new)** Adjust the models complexity according to the newly assigned data.
   (c) Retrain the models using the Expectation-Maximization (EM) algorithm and the segmentation from step (a).
   (d) Select the cluster pair with the largest merge score (based on ΔBIC) that is > 0.0.
   (e) If no such pair of clusters is found, stop and output the current clustering.
   (f) Merge the pair of clusters found in step (c). The models for the individual clusters in the pair are replaced by a single, combined model.
   (g) Go to step (a).

For the merging and clustering stopping criteria, we use a variation of the commonly used Bayesian Information Criterion (BIC) [2]. The ΔBIC compares two possible models: two clusters belonging to the same speaker or to different speakers. The variation used was introduced by Ajmera et al. [4], [6], and consists of the elimination of the tunable parameter λ by ensuring that, for any given ΔBIC comparison, the difference between the number of free parameters in both models is zero.

Both the estimation of the initial number of clusters and the model complexity selection ensure that each individual show starts at an optimum number of clusters, and that each cluster is able to model well the data in it. Although theoretically the initial number of clusters should not be a decisive parameter for an agglomerative clustering system, in practice it turns to be an important factor in the performance of a system. This is probably due to the different resulting number of Gaussian mixtures that are used to model each cluster at each stage of the process. It is therefore important to determine a tradeoff between the number of Gaussian mixtures assigned to each cluster and the number of initial clusters. In sections 3 and 4, we present the relationship between both parameters through a newly defined parameter called the Cluster complexity Ratio (CCR).

3 Model Complexity Selection

The acoustic models used to represent each cluster are a key part of the agglomerative clustering process. On the one hand, comparing their likelihood given

the data is how we decide whether two models belong to the same cluster or not. On the other hand, they are used in the decoding process to redistribute the acoustic data into the different clusters on every iteration.

When designing their size, an important decision is whether we use fixed models (meaning a fixed number of Gaussian mixtures from start to finish), or if we allow the number of Gaussian mixtures to vary according to time or occupancy. Using fixed models is a viable alternative, but runs into the problem of having sufficient training data when the we set the number of Gaussian mixtures to be high, or being too general a model when it is set to be small.

Furthermore, when comparing two models via BIC, if they are too general they tend to over-merge, and when they are too specific to the data they under-merge. Therefore it is important to find a tradeoff on the number of mixtures used (model complexity). This has been addressed in our past systems ([5] and [3]) by using variable complexities as the merging process progresses. In such systems, all cluster models (regardless of their size) are initially trained using a fixed number of Gaussian mixtures. Upon merging any two clusters, the data from both original clusters are merged and a new cluster model is created as the sum of both parents' Gaussian mixtures. This is a variable complexity approach that changes over time.

Such an approach has a drawback that is addressed with our proposed technique. Even though when we start the algorithm, we have the same amount of data assigned to each individual cluster (due to using linear initialization of the available data into clusters), when iterating over decoding the data with the models and merging the different models we obtain clusters with much less data assigned to them that are still modeled with the same complexity than much more populated ones. When performing a BIC comparison, we are comparing more specific models to more general ones, suffering in system performance.

We present an algorithm that selects the number of mixtures to be used when modeling each cluster according to its occupancy. This could be referred to as an **occupancy driven approach**. After each change in the amount of data assigned to each cluster (due to a segmentation), we count the number of acoustic frames that are assigned to each of the models and determine the number of mixtures by:

$$M_i^j = round(\frac{N_i^j}{CCR_{gauss}}) \tag{1}$$

The number of Gaussian mixtures to model cluster i at iteration j (M_i^j) is determined by the number of frames belonging to that cluster at that time (N_i^j) divided by a constant value (CCR_{gauss}) that we call Cluster Complexity Ratio, fixed across all meetings.

In both approaches (time and complexity driven), the total number of mixtures used over all models remains constant in average, being distributed between the different cluster models as described above. This allows tracking of the system evolution by inspection of the Viterbi decoding total likelihood, which can be compared across merging iterations.

When the complexity of any given model changes, we update the model in one of two alternative ways: a) when the final complexity is greater than the initial one, we iteratively split the Gaussian with the biggest posterior probability into two (in the same way as performed within HTK) until the desired number is reached; or b) when the final complexity is smaller than the initial, we erase the initial model and obtain a new model of the given complexity by initializing it to the desired number of Gaussian mixtures given the data.

4 Automatic Selection of the Initial Number of Clusters

In order to perform an agglomerative clustering on the data we need to define an initial number of clusters. This value needs to be higher than the actual number of speakers to allow the system to perform some iterations before finding the optimum number. It also cannot be too big, as each model needs a minimum cluster occupancy to be trained properly, and to avoid unnecessary computation.

In prior work ([5] for the meetings domain and [3] for broadcast news data), the number of initial clusters was fixed within each domain that we work on. In the meetings domain, it was set to either 10 or 16 initial clusters, and in the broadcast news domain it was set to 40 initial clusters. The selection of these values has to be tuned to be greater than the possible number of speakers in any given recording while maximizing the performance.

With the following method, we can estimate the number of initial clusters on a per recording basis by taking into account the total amount of data available for clustering:

$$K = \frac{N_{total}}{G_{clus} CCR_{gauss}} \tag{2}$$

We make the number of initial clusters a function of the amount of data available for clustering, the number of Gaussian mixtures we want to assign per cluster (we use, as in prior work, $G_{clus} = 5$) and the Cluster Complexity Ratio CCR_{gauss}. This initializes the system using an average complexity of G_{clus} and the amount of data per cluster as defined by CCR_{gauss}, which is the same as when defining the models complexity during the previously presented algorithm. This technique does not try to guess the real number of speakers present in a recording, but rather sets an upper boundary to the number of speakers that is closely coupled with the complexity selection algorithm and which allows a correct modeling of each initial cluster for each particular recording.

5 Acoustic Modeling Without Time Restrictions

Our clustering algorithm models the acoustic data using an ergodic hidden Markov model (HMM), where state corresponds to one of the initial clusters. Upon completion of the algorithm's execution, each remaining state is considered to represent a different speaker. Each state contains a set of MD sub-states,

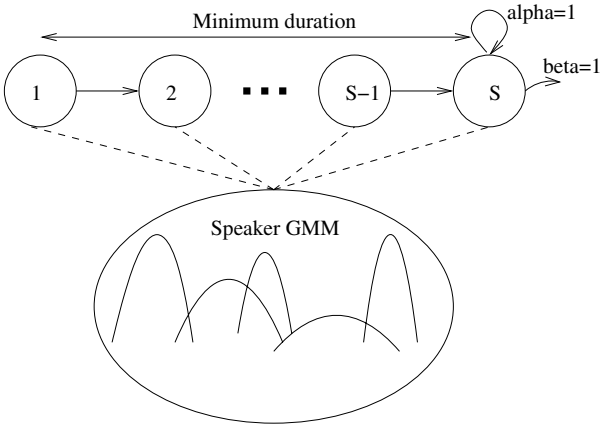


Fig. 1. Cluster models with Minimum duration and modified probabilities

as seen in figure 1, imposing a minimum duration of each model. Each one of the sub-states has a probability density function modeled via a Gaussian mixture model (GMM). The same GMM model is tied to all sub-states in any given state. Upon entering a state, at time n the model forces a jump to the following sub-state with probability 1.0 until the last sub-state is reached. In that sub-state, it can remain in the same sub-state with transition weight α, or jump to the first sub-state of another state with weight β/M, where M is the number of active states/clusters at that time. In prior publications, these were set to $\alpha = 0.9$ and $\beta = 0.1$ (summing to 1).

One disadvantage of using these settings is that it imposes an implicit duration model on the data beyond the minimum duration MD set as a parameter. Such duration modeling changes as we modify the MD value, as illustrated by equations 3 and 4.

$$lkld_{AA} = prob(x(0)|\Theta_A) \prod_{i=1}^{MD-1} (1 \cdot prob(x(i)|\Theta_A)) \cdot \prod_{i=MD}^{2 \cdot MD-1} (\alpha \cdot prob(x(i)|\Theta_A)) \tag{3}$$

$$lkld_{AB} = prob(x(0)|\Theta_A) \prod_{i=1}^{MD-1} (1 \cdot prob(x(i)|\Theta_A)) \cdot \frac{\beta}{M} prob(x(MD)|\Theta_B) \prod_{i=MD+1}^{2 \cdot MD-1} (1 \cdot prob(x(i)|\Theta_B)) \tag{4}$$

Equation 3 shows the computed likelihood given $2MD$ acoustic frames and remaining in cluster A during all of them. Equation 4, on the other hand, shows

the total likelihood if we jump to a model B after the initial MD frames. When both models are the same (A=B) it is desired that eq. 3 be greater than eq. 4 or else the possible speaker turn durations would be strongly quantized to the MD duration. In this case, it happens when $\alpha^{MD} > \frac{\beta}{M}$.

Setting the values of $\alpha = 0.9$ and $\beta = 1 - \alpha$ caused long speaker turns to be artificially penalized against turns with the minimum MD frames. In order to remove this effect (since we do not have a priori information on the average turn length of the input data), we propose to set the value of $\alpha = 1.0$ and $\beta = 1.0$. Thus, once a segment exceeds the minimum duration, the HMM state transitions no longer influences the turn length; turn length is solely governed by acoustics. This creates a non-standard (but valid) HMM topology as $\alpha + \beta$ no longer sums to 1.

6 Experiments and Results

Speaker diarization experiments were conducted using the data distributed for the NIST Rich Transcription 2004 and 2005 Spring Meeting Recognition Evaluation, RT04s and RT05s ([7]). This consists of excerpts from multi-party meetings in English collected at six different sites. From each meeting, only an excerpt of 10 to 12 minutes is evaluated. Although a number of distant microphones is available for each meeting, only the most centrally located microphone (as defined my NIST as the SDM channel) was used to test the algorithms presented here. We merged the RT04s development and evaluation data to create a development set (a total of 16 meeting excerpts), used to adjust some of the parameters in the system. The RT05s evaluation data was used to validate the chosen parameters.

The metric used to evaluate the performance of the system is the same as is used in the NIST RT evaluations and is called Diarization Error Rate (DER). It is computed by first finding an optimal one-to-one mapping of reference speaker ID to system output ID and then obtaining the error as the percentage of time that the system assigns the wrong speaker label. The results given below are the time weighted DER average for the development and evaluation sets.

Although hand-made reference files were provided for each of the sets, they are at times inconsistent and therefore not very suitable to test any new algorithm. In fact, it is planned that for the RT06s evaluation systems will be scored using forced aligned reference files rather than hand-made ones. The automatically generated references are obtained by using a speech recognition system that aligns the words uttered by each speaker to the waveform, and therefore outputs the times where each speaker spoke, suitable for speaker diarization. In the present paper, we used a forced alignment generated using ICSI-SRI ASR system (see [8]). The meeting named NIST_20050412_1303 contains a telephone channel whose transcript was not provided; therefore it was not able to be fully aligned and was taken out of the test set, leaving us with 9 meetings.

In order to select the optimum values for the CCR parameters and the number of Gaussian mixtures per cluster, we did a greedy search on the parameter space using the baseline system including the presented variation to the acoustic models. As we did not perform an exhaustive search, the resulting parameters might not be the optimal ones. The chosen values are CCR = 8 seconds/Gaussian and 5 Gaussian mixtures per initial cluster. The use of both parameters to determine the number of initial clusters sets all recordings to a range of clusters from 10 to 16, which in the meetings environment we have seen to work the best in previous publications.

Using the selected parameters, in table 1 we show the Diarization Error Rates of all presented systems, individually and in conjunction with each other, and the baseline system, both for the development data set and the test set.

Table 1. DER for the development and test sets comparing the different proposed systems

System	Development set	Test set
Baseline system	18.38%	14.43 %
Speaker turn with no time restrictions	17.75%	14.49%
Complexity selection	17.23%	**11.68%**
Initial # models selection	17.59%	14.00%
Complexity + # initial models selection	**16.95%**	12.48%

The baseline system is based on the system presented in [5], with model probabilities $\alpha = 0.9$ and $\beta = 0.1$. The second system introduces the change in the acoustic models to avoid the speaker turn length restrictions. All systems after the second one include such modification. The third and fourth systems correspond to each of the model parameter estimation techniques on their own, and the last system contains all of the proposed techniques.

By avoiding the speaker turn length restriction we obtain an improvement on the development set but not in the evaluation set, though it achieves almost the same result. All other systems improve the baseline results to different degrees. The best system on the development set is the one combining the three presented techniques, although the improvement over the baseline is 8% relative, smaller than the improvement obtained by the complexity selection algorithm on the test set, which is a 19% relative. This indicates the viability of the algorithms to be used on unseen recordings as all parameters had been trained using the development set.

By looking at the overall results, we can generalize that although we have the same number of parameters in the system (before we needed to define the number of Gaussian mixtures per cluster and the number of initial clusters, and now the number of mixtures and the CCR), it tunes better to the individual data sources and is more robust to changes in the show length and the structure of the acoustic data (e.g. how long and how often each speaker speaks).

7 Conclusions

In this paper, we presented three techniques for improving the acoustic modelling of a speaker diarization system based on agglomerative clustering. These techniques define the quantity of initial clusters to use, their complexity (number of Gaussian mixtures) and the topology of the models regarding duration constraints. We introduced a new parameter called the Cluster Complexity Ratio (CCR), which was used to define both the number of initial clusters and the cluster complexity, and allows models to adapt to each individual recording according to the amount of data available for clustering and the structure of the content in the recording. We showed an improvement of up to 8% on the development set and 18% on the test set, which ensures the viability of this method to be used on unseen data.

Acknowledgments

We would like to acknowledge the Speaker Diarization group at ICSI for their thoughtful comments and Joe Frankel, Adam Janin and Jose Pardo for their help. This work was done during Xavier Anguera's stay at ICSI within the Spanish visitors program.

References

1. D. Reynolds and P. Torres-Carrasquillo, "Approaches and applications of audio diarization," in *ICASSP'05*, Philadelphia, PA, March 2005, pp. 953–956.
2. S. Shaobing Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Virginia, USA, Feb. 1998.
3. C. Wooters, J. Fung, B. Peskin, and X. Anguera, "Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system," in *Fall 2004 Rich Transcription Workshop (RT04)*, Palisades, NY, November 2004.
4. J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *ASRU'03*, US Virgin Islands, USA, Dec. 2003.
5. X. Anguera, C. Wooters, B. Peskin, and M. Aguilo, "Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system," in *RT05s Meetings Recognition Evaluation*, Edinburgh, Great Brittain, July 2005.
6. J. Ajmera, I. McCowan, and H. Bourlard, "Robust speaker change detection," *IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 649–651, 2004.
7. NIST rich transcription evaluations, website: http://www.nist.gov/speech/tests/rt.
8. A. Stolcke, X. Anguera, K. Boakye, O. Cetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng, "Further progress in meeting recognition: The icsi-sri spring 2005 speech-to-text evaluation system," in *RT05s Meetings Recognition Evaluation*, Edinburgh, Great Brittain, July 2005.

Speaker Diarization for Multi-microphone Meetings Using Only Between-Channel Differences

Jose M. Pardo[1,2], Xavier Anguera[1,3], and Chuck Wooters[1]

[1] International Computer Science Institute, Berkeley CA 94708 USA
[2] Universidad Politécnica de Madrid, 28040 Madrid, Spain
[3]Technical University of Catalonia, Barcelona Spain
{jpardo,xanguera,wooters}@icsi.berkeley.edu

Abstract. We present a method to extract speaker turn segmentation from multiple distant microphones (MDM) using only delay values found via a cross-correlation between the available channels. The method is robust against the number of speakers (which is unknown to the system), the number of channels, and the acoustics of the room. The delays between channels are processed and clustered to obtain a segmentation hypothesis. We have obtained a 31.2% diarization error rate (DER) for the NIST´s RT05s MDM conference room evaluation set. For a MDM subset of NIST´s RT04s development set, we have obtained 36.93% DER and 35.73% DER*. Comparing those results with the ones presented by Ellis and Liu [8], who also used between-channels differences for the same data, we have obtained 43% relative improvement in the error rate.

1 Introduction

There has been extensive research at ICSI in the last few years in the area of speaker segmentation and diarization [1,2,3,4,5,6,7]. Speaker diarization is the task of identifying the number of participants in a meeting and create a list of speech time intervals for each such participant.

The task of speaker diarization for meetings with many speakers and multiple distant microphones (MDM) should be easier compared to the use of a single distant-microphone (SDM) because: a) there are redundant signals (one for each channel) that can be used to enhance the processed signal, even if some of the channels have a very poor signal to noise ratio; and b) there is information encoded in the signals about the spatial position of the source (speaker) that is different from one to another. In previous work [9], a processing technique using the time delay of arrival (TDOA) was applied to the different microphone channels by delaying in time and summing the channels to create an enhanced signal. With this enhanced signal, the speaker diarization error (DER) was improved by 3.3% relative compared to the single channel error for the RT05s evaluation set, 23% relative for the RT04s development set, and 2.3% relative for the RT04s evaluation set (see [10] for more information about the databases and the task).

It is important to emphasize that the task is done without using any knowledge about the number of speakers in the room, their location, the locations and quality of the microphones, or the details of the acoustics of the room.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 257 – 264, 2006.

While in the work mentioned above, improvements were obtained, no direct information about the delays between different microphones was used in the segmentation and clustering process. In order to study and analyze the information contained in the delays, we have performed some experiments to determine to what extent the delays by themselves can be used to segment and cluster the different speakers in a room. We have tried to develop a system that is robust to the changes in the meeting conditions, room, microphones, speakers, etc.

The only work of which we are aware that only uses between-channel differences for speaker turn segmentation is the work of Ellis and Liu [8]. In their work, they used the cross correlation between channels to find a peak that represents a delay value between two channels. They later clustered the delay values to create segments in the speech frames. The results they reported for the set of shows corresponding to the RT04s development set is 62.3% DER* error.[1] We present a method to use only the delays to obtain a segmentation hypothesis. Using our method, we obtain a diarization error (DER*) [10] of 35.73% for the same set of shows. Furthermore, for the set of shows corresponding to the RT05s MDM conference evaluation set, we have obtained a 31.2% DER error. The DER error could be reduced further, since one of the shows had a large number of false alarm speech errors (due to big background noises such as papers rustling, etc). Without taking this show into account, the average DER error rate for the RT05s set goes down to 27.85%.

The paper is organized as follows: In Section 2 we describe the basics of our system, in Section 3 we describe the experiments done, in Section 4 we discuss the results, Section 5 finishes with our conclusions.

2 Description of the System

2.1 Delay Generation

Given any two microphones (i and j) and one source of speech (x[n]), let us call the signals received by each microphone $x_i[n]$and $x_j[n]$.

We define the delay of $x_i[n]$ with respect to $x_j[n]$ as the time difference of the sound arriving at each microphone.

If we assume the produced wave-front is flat when reaching the microphones, and further assume a non-dispersive wave propagation, we obtain the delay (in # of samples) as

$$d(i,j) = \frac{D(i,j)\cos\alpha}{c.f_s} \tag{1}$$

Where D(i,j) is the distance between the two microphones, α is the angle of arrival of the source speech, c is the speed of sound (in m/sec) and f_s is the sampling frequency (in samples/sec.), see [9].

In order to estimate the TDOA between segments, we cannot directly use equation (1) because we do not know the number of speakers nor their locations. We used a

[1] The equivalent that they used is DER minus False Alarm (in NIST terminology), we called it DER*.

modified version of the Generalized Cross Correlation with phase transform ($GCC_{PHAT}(f)$) (see [11]) and estimate the delays between microphones with the following formula:

$$d(i,j)=^{\arg\max}_{d}(R_{PHAT}(d)) \tag{2}$$

$R_{PHAT}(d)$ is the inverse transform of $G_{PHAT}(f)$ (the generalized cross correlation)

For a set of microphones, we choose any microphone as the reference microphone and calculate the delay of the signals coming to the other microphones relative to the reference microphone. We form a vector of these delays that has as many components as the number of microphones minus 1. We use a window width of 500 msec with a shift of 10 msec per frame. Non-speech frames are estimated with the SRI Meetings speech/non-speech detector and are excluded from the subsequent process see [5]. All the data given below about speech/non-speech errors exclusively originate from this system.

2.2 Segmentation and Agglomerative Clustering

The segmentation and clustering is very similar to what is proposed in [3] for segmentation and clustering using acoustic features. We use the vectors explained above to feed the initial segmentation and posterior resegmentation and clustering as proposed in [3]. Essentially, the process consists of two modules: the initialization and the clustering. The initialization requires a "guess" at the maximum number of speakers (K) that are likely to occur in the data. The data are then divided into K equal-length segments, and each segment is assigned to one model. Each model's parameters are then trained using its assigned data. To model each cluster, we use and HMM consisting of a minimum number of states all with the same output pdf -a gaussian mixture- with a diagonal covariance matrix starting with "g" gaussians per model. These are the models that seed the clustering and segmentation processes described next.

Merging Score

One of the main problems in the segmentation and clustering process is deciding which merging score to use. The BIC criterion has been used extensively, giving good results [1,12] and the modification of BIC to eliminate the need of a penalty term that compensates for different number of parameters has given us also good results 3], although it is an important open question how much it depends on the kind of data vectors and models that are used in the comparisons.

The modified BIC is the following:

$$\log p(D/\theta) \geq \log p(D_a/\theta_a) + \log p(D_b/\theta_b) \tag{3}$$

θ_a is the model created with D_a and θ_b is the model created with D_b

θ is the model created with D, with the number of parameters in θ equal to the sum of the number of parameters in θ_a plus the number of parameters in θ_b

Clustering Process
The iterative segmentation and merging process consists of the following steps:

1. Run a Viterbi decode to re-segment the data.
2. Retrain the models using the segmentation from (1).
3. Select the pair of clusters with the largest merge score (Eq. 3)> 0.0 (Since Eq. 3 produces positive scores for models that are similar, and negative scores for models that are different, a natural threshold for the system is 0.0).
4. If no pair of clusters is found, stop.
5. Merge the pair of clusters found in (3). The models for the individual clusters in the pair are replaced by a single, combined model.
6. Go to (1).

3 Experiments and Evaluation

We have used the RT05s MDM conference meetings evaluation data in our initial development experiments. The data consists of 10 meetings from which 10 minutes excerpts for every one have been extracted [10]. Several combinations of the parameters "g" and "K" have been tried, with the best results obtained using 1 mixture and 10 initial clusters. The speaker diarization errors obtained with several combinations of these parameters are presented in Table 1.

Table 1. Speaker diarization errors DER for the RT05s MDM conference room eval set

	# of initial clusters	
# of gaussians	**10**	***20***
1	31.20 %	34.77 %
2	38.68%	43.49%

The breakdown of these data (1 gaussian, 10 initial clusters) into different shows is presented in Table 2. We show the Missed Speech error, the False Alarm Speech error, the Speech/NonSpeech error (SpNsp), the Speaker error and the overall DER error[2]. In the results presented, the regions where more than one speaker are talking have been excluded[3]. We have analyzed the results and found that the show VT_20050318-1430 has a big SpNsp error, and particularly the False Alarm error. This is due to a background paper noise that is erroneously detected as speech by the SRI system. Without taking into account this show, the average DER is 27.85%. We have also investigated the minimum error (ORACLE) that could be obtained by this procedure by using the clustering iteration loop without any stopping criterion and calculating the theoretical error obtained if the system stopped after each iteration.

[2] The speech/non-speech segmentation is not calculated by our system and it is presented here for completeness.
[3] This condition is considered in the evaluation tool as the "no overlap" condition.

The DER (ORACLE) error for these shows (not including VT_20050318-1430) is 23.28%. This error is just a way of measuring the possible absolute limit for our current experiments if an optimum stopping criterion were known.

Table 2. Missed speech, False Alarm speech, Speech/Non-speech error and Diarization error for the RT05s eval set using 1 mixture and 10 initial clusters

File	Miss	FA	SpNsp	Spkr	Total
AMI_20041210-1052	1,1	1,9	3	13,5	16,53
AMI_20050204-1206	1,8	1,7	3,5	19,6	23,03
CMU_20050228-1615	0,1	1	1,1	17,2	18,28
CMU_20050301-1415	0,2	3,3	3,5	42,4	45,88
ICSI_20010531-1030	4,3	1,3	5,6	15	20,59
ICSI_20011113-1100	2,9	2,7	5,6	39,9	45,52
NIST_20050412-1303	0,6	2,9	3,5	21,7	25,19
NIST_20050427-0939	1,5	2,5	4	33,2	37,18
VT_20050304-1300	0	3,6	3,6	22,1	25,7
VT_20050318-1430	0,3	22,6	22,9	38,4	61,27
ALL	1,3	4	5,3	25,9	31,2

For the purpose of comparison of this method with the regular method, we have run the same standard procedure but now using the normal MFCC feature set and we obtained a DER error of 13.38% for the same set of shows (not including VT_20050318-1430 for the reasons mentioned above)[4]. This data shows us that there is still a big gap between the errors obtained using only time differences and the ones obtained using only acoustic data.

Table 3. Comparisons between results obtained by Ellis and Liu and our results in the same subset of shows from NIST RT04 development data

Meeting	Ellis DER*	Our System DER*	Our System DER	Number of microphones used
LDC_20011116-1400	66%	6.89%	8.89%	4
LDC_20011116-1500	77.3%	59.33%	59.63%	4
NIST_20020214-1148	58%	33.32%	37.72%	4
NIST_20020305-1007	46.1%	32.81%	34.11%	4
ICSI_20010208-1430	49.1%	29.9%	38.7%	4
ICSI_20010322-1450	63.3%	43.53%	43.83%	4
Average All	**62.3%**	**35.73%**	**36.93%**	

[4] It should be noted however that these results have been obtained using the Standard system presented at the NIST RT05s meeting BUT without the purification system included[3].

To further demonstrate that there is information in the timing differences between channels, we ran an experiment using just random numbers and processed them to extract the diarization error. For the show processed in this manner, ICSI_20010531-1030 we obtained 93.23% DER error compared to 21.14% DER error using the same parameters settings. The system is able to find information in the time differences between signals coming from different microphones.

In order to be able to compare our results with the ones presented by Ellis and Liu [8], we have run the system with the same set of shows that they used in their experiments, and reducing the number of channels to 4 in all cases. In Table 3, the comparisons of both experiments are presented. It is important to notice that in these results, two of the shows from NIST RT04 (the CMU shows) have not been used because the conditions of these shows (only one distant microphone) are not compatible with the conditions of our experiment (multiple distant microphones)[5]. Also the results presented here include the overlap regions and no False Alarms (we call it the DER* error). We have included in Table 3 also the standard DER error for completeness. The analysis of the results show a big improvement of our system compared to the Ellis one. The differences may well come from the different way we use to calculate the delays between signals and the different segmentation and clustering procedure. Since the number of microphones used in this experiment were less than the number of microphones available, we have also investigated the error rate that we could obtain for the same set of shows if we used all the available microphones. Table 4 gives results of this comparison. It can be seen that the use of more microphones reduces the DER error rate by 3.26% absolute.

Table 4. Comparisons between DER rates obtained using 4 channels and results using all the channels available in the system

Meeting	# microphones used	Diarization error	# microphones used	Diarization error
LDC_20011116-1400	4	8.89%	7	12.26%
LDC_20011116-1500	4	59.63%	8	45.72%
NIST_20020214-1148	4	37.72%	7	36.40%
NIST_20020305-1007	4	34.11%	6	41.37%
ICSI_20010208-1430	4	38.7%	6	19.81%
ICSI_20010322-1450	4	43.83%	6	44.68%
Average All		**36.93%**		**33.67%**

4 Discussion

The estimation of errors in the Ellis system was performed with a quantized version of the scoring method that we have used (the official NIST scoring program). In his

[5] Ellis and Liu developed an artificial condition for those two shows that do not make sense in our method. Those two shows are then not used.

scoring the errors were quantized in segments of length 250 msec and no reference was made to the forgiveness collar of 250 msec at each side of a reference segment that was done in the NIST scoring software. Also, from the explanations given in their paper, they did not count the regions of silence in the reference transcriptions. We have discounted those errors in our data and defined DER* (see column 3 of Table 3).

If we compare our results to Ellis results, there is an important improvement. Our experiments further support the Ellis and Liu idea that there is information in the timing differences between different channels that can be used to extract speaker turn information (obvious in any case but usually difficult to extract). However if we compare the results that we obtain with the results obtained with our standard spectral system there is still a big gap to cover. Nonetheless, in this paper we just wanted to show that there is information in the timing differences between channels that could be used in speaker diarization systems. It is our purpose to continue research in this area in order to be able to integrate information coming from different sources and apply it to this task.

5 Conclusions

In this paper we have presented some experiments to analyze the information that exists in the timing differences between channels in the speaker diarization task for multiple distant microphones. While our results are significantly better than the ones published up to now with the same type of information, these results should be considered as a first step towards the development of improved systems for speaker diarization in the presence of multiple microphones.

Acknowledgements

This work was supported by the Joint Spain-ICSI Visitor Program. We also would like to thank Andreas Stolcke, Kemal Sönmez and Nikki Mirghafori for many helpful discussions. We acknowledge the help of Adam Janin in reviewing the paper.

References

1. J. Ferreiros, D. Ellis: Using Acoustic Condition Clustering To Improve Acoustic Change Detection On Broadcast News. Proc. ICSLP 2000
2. J. Ajmera, C,Wooters : A Robust speaker clustering algorithm, IEEE ASRU 2003.
3. X. Anguera, C. Wooters, B. Pesking and Mateu Aguiló : Robust Speaker Segmentation for Meetings: The ICSI-SRI Spring 2005 Diarization System, Proc NIST MLMI Meeting Recognition Workshop, Edinburgh, 2005
4. C. Wooters, N. Mirghafori, A. Stolcke, T. Pirinen, I Bulyko, D. Gelbart, M. Graciarena, S. Otterson, B. Peskin and M. Ostendorf :"The 2004 ICSI-SRI-UW Meeting Recognition System" In Proceedings of the Joint AMI/Pascal/IM2/M4 Workshop on Meeting Recognition. Also published in Lecture Notes in Computer Science, Volume 3361 / 2005.
5. C.Wooters, J, Fung, B. Pesking, X. Anguera, “Towards Robust Speaker Segmentation: The ICSI-SRI Fall 2004 Diarization System” NIST RT-04F Workshop, Nov. 2004.

6. A. Stolcke, X. Anguera, K. Boakye, O. Cetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters and J. Zheng, "Further Progress in Meeting Recognition: The ICSI-SRI Spring 2005 Speech-to-Text Evaluation System" Proceedings of NIST MLMI Meeting Recognition Workshop, Edinburgh.
7. A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, J. Macias-Guarasa, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters and B. Wrede, "The ICSI Meeting Project: Resources and Research" NIST ICASSP 2004 Meeting Recognition Workshop, Montreal
8. D.P.W Elis and Jerry C.Liu : Speaker Turn Segmentation Based On Between-Channels Differences, Proc. ICASSP 2004.
9. X. Anguera, C. Wooters, J. Hernando : Speaker Diarization For Multi-Party Meetings Using Acoustic Fusion, IEEE ASRU, 2005.
10. NIST Spring 2005 (RT05S) Rich Transcription Meeting Recognition Evaluation Plan, http://www.itl.nist.gov/iad/894.01/tests/rt/rt2005/spring/
11. M.S. Brandstein and H.F. Silverman: A Robust Method For Speech Signal Time-Delay Estimation In Reverberant Rooms, ICASSP 97, Munich
12. S.S. Chen, P.S. Gopalakrishnan: Speaker Environment And Channel Change Detection And Clustering Via The Bayesian Information Criterion, Proceedings DARPA Broadcast News Transcription and Understanding Workshop, Virginia, USA, Feb. 1998

Warped and Warped-Twice MVDR Spectral Estimation With and Without Filterbanks

Matthias Wölfel

Institut für Theoretische Informatik, Universität Karlsruhe (TH)
Am Fasanengarten 5, 76131 Karlsruhe, Germany
wolfel@ira.uka.de
http://isl.ira.uka.de/~wolfel

Abstract. This paper describes a novel extension to warped *minimum variance distortionless response* (MVDR) spectral estimation which allows to steer the resolution of the spectral envelope estimation to lower or higher frequencies while keeping the overall resolution of the estimate and the frequency axis fixed. This effect can be achieved by the introduction of a second bilinear transformation to the warped MVDR spectral estimation, but now in the frequency domain as opposed to the first bilinear transformation which is applied in the time domain, and a compensation step to adjust for the pre-emphasis of both bilinear transformations. In the feature extraction process of an automatic speech recognition system this novel extension allows to emphasize classification relevant characteristics while dropping classification irrelevant characteristics of speech features according to the characteristics of the signal to analyze.

We have compared the novel extension to warped MVDR and the traditional *Mel frequency cepstral coefficients* (MFCC) on development and evaluation data of the Rich Transcription 2005 Spring Meeting Recognition Evaluation lecture meeting task. The results are promising and we are going to use the described warped and warped-twice front-end settings in the upcoming NIST evaluation.

1 Introduction

To improve phoneme classification, and therefore word accuracy, it is important to emphasize classification relevant characteristics while dropping the irrelevant characteristics in the feature extraction process of an automatic speech recognition system. Traditionally this is achieved by successive implementations (e.g, a spectral envelope or/and a filterbank followed by cepstral transformation, cepstral normalization and linear discriminant analysis) treating all phoneme types equally. As for different phoneme types the important regions on the frequency axis vary [1], e.g. low frequencies for vowels and high frequencies for fricatives, it is a natural extension to the traditional approach to vary the spectral resolution depending on the phoneme/signal to analyse. To provide a framework to allow for these adjustments we propose an extension to the warped *minimum variance distortionless response* (MVDR) by a second bilinear transformation. This novel spectral envelope estimate has two ways of freedom to control spectral resolution. The first is the number of linear prediction coefficients also referred to as

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 265–274, 2006.

model order. E.g. increasing the model order for the underlying linear parametric model is also increasing the overall spectral resolution and vice versa. In previous work [2] we have demonstrated that a speaker dependent model order can improve speech recognition. The second, novel way of freedom, is a compensated warp factor which allows to steer spectral resolution to lower or higher frequency regions without changing the frequency axis.

2 Warped-Twice MVDR Spectral Envelope Estimation

The use of MVDR as a spectral envelope technique was previously proposed by Murthi and Rao [3,4] and applied to speech recognition by Dharanipragada and Rao [5]. Moreover, to ensure that more parameters in the spectral model are allocated to the low, as opposed to the high, frequency regions of the spectrum, thereby mimicking the frequency resolution of the human auditory system, we have extended this approach by *warping* the frequency axis with the bilinear transformation prior to MVDR spectral estimation [6,2], therefore dubbed *warped MVDR*. To allow to bend the frequency axis without a change of resolution, a second bilinear transformation is introduced into the warped MVDR framework, yet applied in the frequency domain rather than in the time domain. To see the differences between the bilinear transformation applied in the time or frequency domain we have to understand that the warping in the time

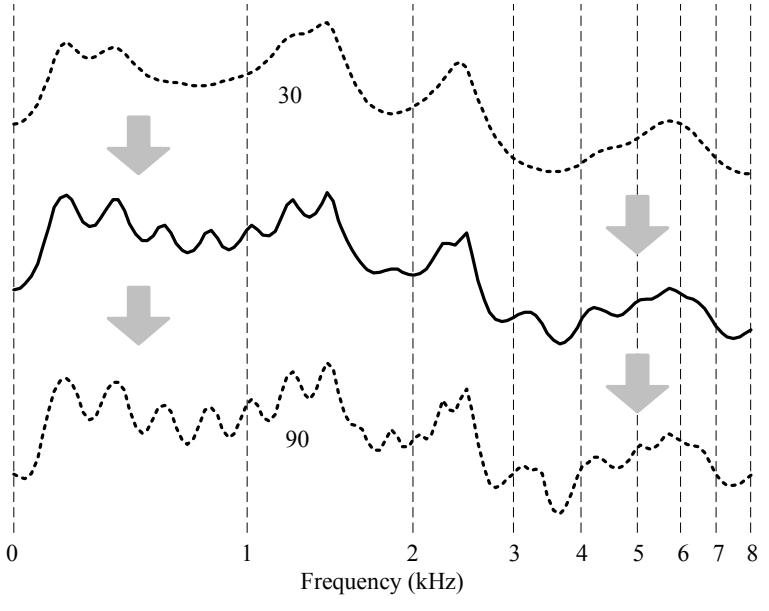


Fig. 1. *Influence of model order:* The solid line shows an envelope with the model order 60 and the time domain warp 0.4595 and its counterparts with lower and higher model order as dashed lines. The arrows are pointing in the direction of higher resolution.

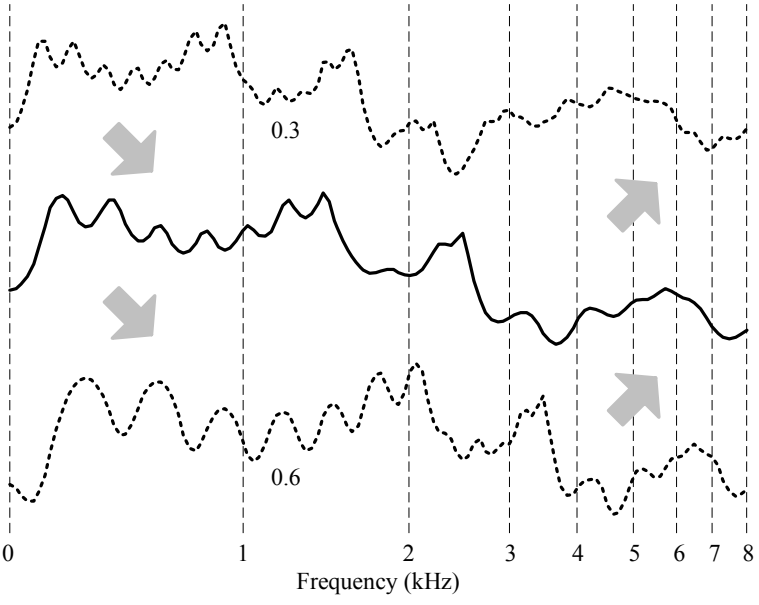


Fig. 2. *Influence of time domain warp:* The solid line shows an envelope with the model order 60 and the time domain warp 0.4595 and its counterparts with lower and higher time domain warp factor as dashed lines. The arrows are pointing in the direction of higher resolution.

domain moves more coefficients to lower or higher frequencies *before* spectral analysis and therefore resulting in an increase or decrease of resolution in lower or higher frequency regions. While warping in the frequency domain is a post process *after* spectral analysis leaving the coefficients untouched and therefore also the frequency resolution. A new aspect comes into play if the warp factor of the bilinear transformation in the frequency domain is set to compensate for the bending of the frequency axis introduced by the bilinear transformation in the time domain [7]. This allows to keep the frequency axis fixed at a particular warped axis, like the Mel-frequency, while moving the spectral resolution to low or high frequency regions. Due to the application of two bilinear transformations we have dubbed this approach *warped-twice MVDR* (W2MVDR).

For a better understanding we want to have a closer look at the different degrees of freedom, namely the number of *linear prediction coefficients* (LPC)s also referred to as model order, the warping in the time domain as well as in the frequency domain and the compensation of the time domain warp by the frequency domain warp. An increase of the model order, Figure 1, increases the resolution of the spectral estimate equally over all frequencies while a decrease lowers the overall resolution of the spectral estimate. The warp factor applied in the time domain, Figure 2, bends the frequency axis and changes the resolution of the spectral envelope. Therefore, it can be used to apply the Mel-frequency. Let

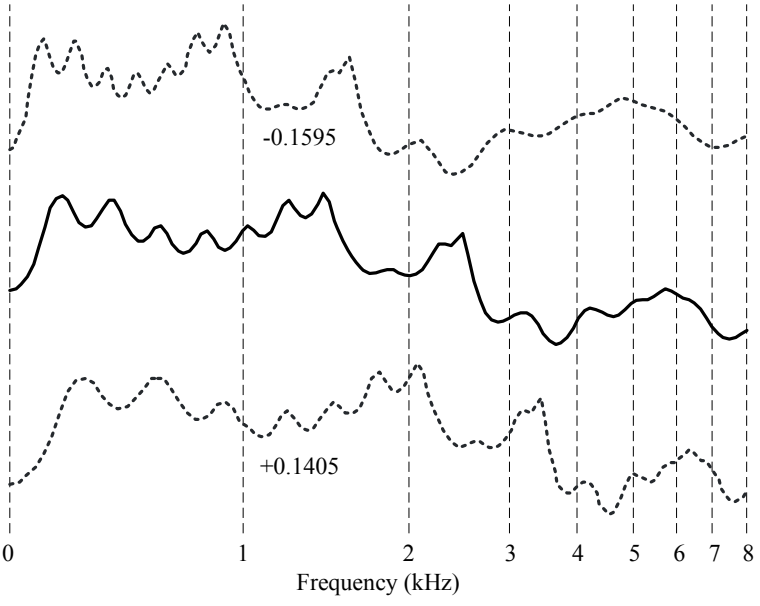


Fig. 3. *Influence of frequency domain warp:* The solid line shows an envelope with the model order 60 and the time domain warp 0.4595 and its counterparts with lower and higher frequency domain warp factor as dashed lines. No change of resolution has taken place.

us consider a warp factor smaller than the Mel-frequency $\alpha < \alpha_{\text{Mel}}$, we observe that the spectral resolution improves in high frequency regions and decreases in low frequency regions and vice versa. On the other hand, applying the warp in the frequency domain, Figure 3, bends the frequency axis without changing the resolution. This could be used to implement vocal tract length normalization (not used in our experiments as the traditional approach of piece-wise linear warping is leading to better results). Finally, let us consider the case where the warp in time domain is compensated by a warp in frequency domain, Figure 4. In this case the frequency axis is constant, but an increase of spectral resolution can be observed in high frequency regions for a warp value smaller than the Mel-frequency $\alpha < \alpha_{\text{Mel}}$ and a decrease in low frequency regions and vice versa.

2.1 Fast Computation

For a fast computation of the W2MVDR envelope we have extended Musicus' [8] algorithm to calculate the MVDR envelope of model order N from the LPC $a^{(N)}_{0\cdots N}$ of model order N as follows:

1. **Computation of the warped autocorrelation coefficients** $\tilde{R}_{0\cdots N+1}$
   To derive warped autocorrelation coefficients, the linear frequency axis ω has to be transformed to a warped frequency axis $\tilde{\omega}$ by replacing the unit delay element z^{-1} with a *bilinear transformation*

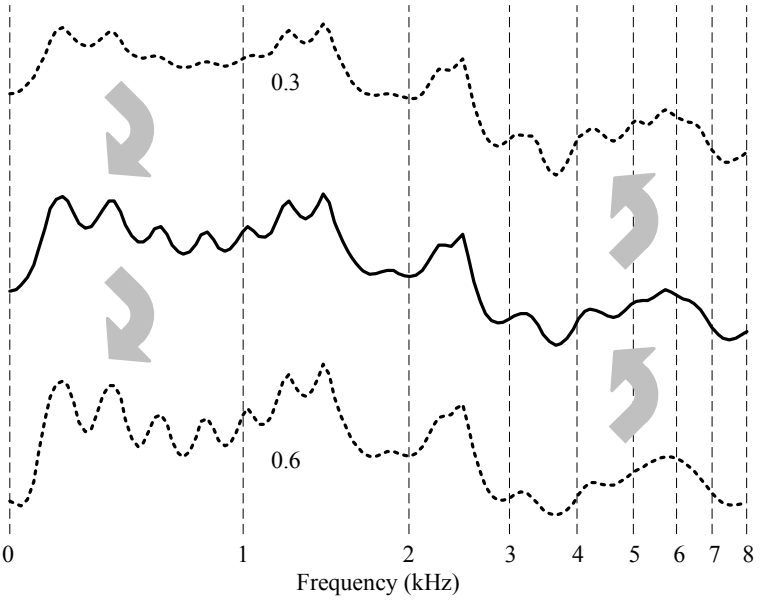


Fig. 4. *Influence of frequency compensated time domain warp:* The solid line shows an envelope with the model order 60 and the time domain warp 0.4595 and its counterparts with lower and higher time domain warp with compensation in the frequency domain to fit a final warp factor of 0.4595 as dashed lines. The arrows are pointing in the direction of higher resolution.

$$z = \frac{z^{-1} - \alpha}{1 - \alpha \cdot z^{-1}} \tag{1}$$

Therefore we can derive the warped autocorrelation coefficients by

$$\tilde{R}[m] = \sum_{n=0}^{\infty} \tilde{x}[n]\tilde{x}[n-m] \tag{2}$$

where the frequency-warped speech signal $\tilde{x}$ is defined by

$$\tilde{X}(\tilde{z}) = \sum_{n=0}^{\infty} \tilde{x}[n]\tilde{z}^{-n} = \sum_{n=0}^{N-1} x[n]z^{-n} \tag{3}$$

A direct calculation of warped autocorrelation coefficients is not feasible, because a bilinear transformed *finite* sequence results in an *infinite* sequence and therefore is approximated [9].

2. **Calculation of the compensation warp parameter**
   To fit the final frequency axis to the Mel-frequency α_{Mel} we have to compensate the first warp α by a second warp:

$$\beta = \frac{\alpha - \alpha_{\text{Mel}}}{1 - \alpha \cdot \alpha_{\text{Mel}}} \tag{4}$$

   For a signal sampled at 16 kHz α_{Mel} has to be 0.4595.

3. **Compensation of the pre-emphasis**
To derive the distortion introduced by the concatenated bilinear transformations with warp values α and β we calculate the phase delay by a frequency derivative of one bilinear transformation (1) with the warp value

$$\chi = \frac{\alpha + \beta}{1 + \alpha \cdot \beta}$$

and express the result in the *weighting function*:

$$\left|\tilde{W}(\tilde{z})\right|^2 = \frac{1 - \chi^2}{(1 + \chi \cdot \tilde{z}^{-1})^2} \tag{5}$$

This is a pre-emphasis filter causing the spectrum at the output to be not perfectly flat. To compensate for this unwanted effect, to get a flat spectrum, we have to apply the inverted weighting function

$$\left|\tilde{W}(\tilde{z}) \cdot \tilde{W}(\tilde{z}^{-1})\right|^{-1} = \frac{1 + \chi^2 + \chi \cdot \tilde{z}^{-1} + \chi \cdot \tilde{z}}{1 - \chi^2}$$

to the warped autocorrelation coefficients $\tilde{r}$. This can be realized as a second order finite impulse response filter:

$$\widehat{R}[i] = \frac{1 + \chi^2 + \chi \cdot \tilde{R}[i-1] + \chi \cdot \tilde{R}[i+1]}{1 - \chi^2} \tag{6}$$

where $\tilde{R}[-1] = \tilde{R}[1]$.
The effect of pre-emphasis, and its compensation, on the spectral envelope due to the bilinear-transformations is depicted in Figure 5.

4. **Computation of the warped-LPCs** $\widehat{a}^{(N)}_{0\cdots N}$
The warped-LPCs can now be estimated by the Levinson-Durbin recursion [10] replacing the linear autocorrelation coefficients R with their warped and pre-emphasis compensated counterparts $\widehat{R}$.

5. **Correlation of the warped-LPC**

$$\widehat{\mu}_k = \begin{cases} \sum_{i=0}^{N-k} (N + 1 - k - 2i) \widehat{a}^{(N)}_i \widehat{a}^{*(N)}_{i+k} & : k = 0, \cdots, N \\ \widehat{\mu}^*_{-k} & : k = -N, \cdots, -1 \end{cases}$$

6. **Computation of the W2MVDR envelope**

$$S_{\mathrm{W2MVDR}}(\omega) = \frac{\epsilon}{\sum_{k=-N}^{N} \widehat{\mu}_k \frac{e^{j\omega} - \beta}{1 - \beta \cdot e^{j\omega}}} \tag{7}$$

ϵ: inverse of the prediction error variance.

Note that (7) is already in the Mel-warped frequency domain and therefore we have to replace the Mel-filterbank in the front-end of a speech recognition system by a filterbank of uniformly half overlapping triangular filters.

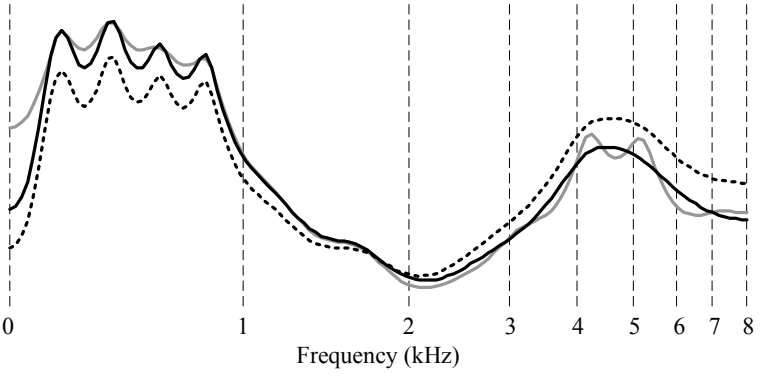


Fig. 5. The gray line — the reference — shows an envelope with model order 30 and warp factor 0.4595. The black lines are spectral envelopes with a warp factor of 0.6595 and same model order as before. The pre-emphasis is not compensated at the dotted line while on the solid line it is. It is clear to see that the solid line is following the amplitude of the gray line while the dotted line is emphasizing high frequencies.

7. **Scaling of the W2MVDR envelope**
   With the relation $\log(a + b) \approx \log(\max\{a, b\})$ we can conclude that *spectral peaks* are particular robust to additive noise in the logarithmic domain. Therefore, to get features which are less distorted by additive noise we match the W2MVDR envelope to the highest spectral peak of the Fourier spectrum [2].

The warped MVDR spectral estimation is a special case of the W2MVDR where $\alpha = \alpha_{\text{Mel}}$. Our previous definition of the warped MVDR is without compensation of the pre-emphasis. Through experiments we found that a compensation of the pre-emphasis results in an additional gain of 0.1% in word accuracy on the third pass but might lose on previous passes.

3 Steering Function

As we wish to adapt our spectral envelope estimate by the free parameters of the W2MVDR envelope, we have to find a steering function in such a way that classification relevant characteristics are emphasized, while less relevant information is suppressed. One promising approach to steer the spectral resolution to lower or higher frequencies was suggested in the work by Nakatoh et al. [7]. There, for every frame indexed i, a division of the first $R[1]$ by the zero $R[0]$ autocorrelation coefficient was used

$$0 \leq \varphi_i = \left| \frac{R_i[1]}{R_i[0]} \right| \leq 1 \tag{8}$$

In combination with γ to adjust the sensibility to the normalized autocorrelation coefficient and the subtraction of the bias to keep the average of α close to α_{Mel} we can write

$$\alpha_i = \gamma \cdot (\varphi_i - 0.5) + \alpha_{\text{Mel}} \tag{9}$$

which is a slight modification to the original proposal by Nakatoh et al. For our experiments we kept γ fix at 0.1.

4 Speech Recognition Experiments

In order to evaluate the performance of the proposed W2MVDR spectral estimation in combination with the steering factor we ran experiments on close talking development and evaluation data of the Rich Transcription 2005 Spring Meeting Recognition Evaluation [11] lecture meeting task.

As a speech recognition engine we have used the *Janus Recognition Toolkit* (JRTk), which is developed and maintained by the Interactive Systems Laboratories at two sites: Universität Karlsruhe (TH), Germany and Carnegie Mellon University, USA. Relatively little supervised in domain data is available for acoustic modeling of the recordings. Therefore, we decided to train the acoustic model on the close talking channel of meeting corpora and merge it with the *Translanguage English Database* (TED) corpus [12] summing up to a total of approximately 100 hours of training material. The front-end provided features every 10 ms (1. and 2. pass) or 8 ms (3. pass) obtained by the Fourier transformation, warped MVDR or W2MVDR spectral estimation followed by a Mel (Fourier), linear (MVDR, high MO) or no filterbank (MVDR, low MO) and a discrete cosine transformation. Thereafter 13 or 20 cepstral coefficients were mean and variance normalized and after taking 7 adjacent reduced to 42 dimensions with linear discriminant analysis. The acoustic model after merge and split training consisted of approximately 3,500 context dependent codebooks with up to 64 Gaussians with diagonal covariances each, summing up to a total of approximately 180,000 Gaussians. To train a 3-gram language model we have used corpora consisting of broadcast news, proceedings of conferences such as ICSLP, Eurospeech, ICASSP, ACL and ASRU and talks by the Translanguage English Database. The vocabulary contains approximately 23,000 words, the perplexity is around 125 with an out of vocabulary rate below 1.5%.

The *word error rates* (WER)s of our speech recognition experiments for different spectral estimation techniques and passes are shown in Table 1. The first pass is unadapted while the second and third passes are adapted on the hypothesis of the previous pass using *maximum likelihood linear regression* (MLLR), feature space adaptation (constrained MLLR) and vocal track length normalization.

Comparing the W2MVDR front-end with model order 60 and filterbanks to its warped MVDR counterpart we observe a constant gain of at least 0.5% in word accuracy. If we wish to neclect the filterbank, we have to compensate for its smoothing behaviour by reducing the model order to 30 and — for best performance — we have to increase the number of cepstral coefficients to 20. This leads to an improvement of at least 0.6% in word accuracy compared to the warped MVDR with filterbanks. Even though the W2MVDR front-end without

filterbanks is able to beat the performance of all other investigated front-ends on the third pass of Dev05, we can conclude that no further impovement in performance over the warped MVDR without filterbanks and increased number of cepstral coefficients can be reached by the W2MVDR front-end without filterbanks.

Table 1. *Word error rates* (WER)s for different front-end types and settings, FB: filterbank, MO: model order, CC: number of cepstral coefficients

Spectrum	FB	MO	CC	WER					
Test Set				Dev05			Eval05		
Pass				1	2	3	1	2	3
Fourier	Y	–	13	36.1%	30.3%	28.0%	35.3%	29.7%	27.7%
warped MVDR	Y	60	13	35.0%	30.0%	28.2%	35.5%	29.9%	27.6%
W2MVDR	Y	60	13	34.5%	29.5%	27.5%	34.1%	29.2%	27.0%
warped MVDR	N	30	13	34.6%	29.8%	27.8%	34.7%	29.6%	27.2%
warped MVDR	N	30	20	33.9%	29.1%	27.4%	34.9%	29.2%	26.9%
W2MVDR	N	30	20	34.7%	29.3%	27.1%	35.4%	29.5%	27.1%

5 Conclusion

We have investigated warped and warped-twice MVDR spectral estimation on the Rich Transcription 2005 Spring Meeting Recognition lecture meeting task. We found that MVDR front-ends are in general superior to Fourier based front-ends. Furthermore, we have demonstrated that the W2MVDR is superior to the warped MVDR spectral envelope if followed by filterbanks. Neglecting the filterbanks we have observed that an increased number of cepstral coefficients leads to a better accuracy. In this case the W2MVDR can not further improve the performance over the warped MVDR.

From preliminary studies on confusion network combination on the development data gaining more than 2% in accuracy on top of the best single system, we hope to see further improvements by the investigation of cross adaptation in combination with confusion network combination, where we will in particular focus on W2MVDR with high model orders and filterbanks in combination with warped MVDR with low model order and neglected filterbanks.

Acknowledgment

The work presented here was partly funded by the *European Union* (EU) under the integrated project CHIL, Computers In the Human Interaction Loop (Grant number IST-506909).

References

1. N. Malayath, "Data-driven methods for extracting features from speech," Ph.D. dissertation, Oregon Graduate Institute of Science and Technology, January 2000.
2. M. Wölfel and J. McDonough, "Minimum variance distortionless response spectral estimation, review and refinements," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, Sept. 2005.
3. M. Murthi and B. Rao, "All-pole model parameter estimation for voiced speech," *IEEE Workshop Speech Coding Telecommunications Proc., Pacono Manor, PA*, 1997.
4. ——, "All-pole modeling of speech based on the minimum variance distortionless response spectrum," *IEEE Trans. Speech Audio Processing*, vol. 8, no. 3, pp. 221–239, May 2000.
5. S. Dharanipragada and B. Rao, "MVDR based feature extraction for robust speech recognition," *Proc. ICASSP*, vol. 1, pp. 309–312, 2001.
6. M. Wölfel, J. McDonough, and A. Waibel, "Minimum variance distortionless response on a warped frequency scale," *Proc. Eurospeech*, pp. 1021–1024, 2003.
7. Y. Nakatoh, M. Nishizaki, S. Yoshizawa, and M. Yamada, "An adaptive Mel-LP analysis for speech recognition," *Proc. ICSLP*, 2004.
8. B. Musicus, "Fast MLM power spectrum estimation from uniformly spaced correlations," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 33, pp. 1333–1335, 1985.
9. H. Matsumoto and M. Moroto, "Evaluation of Mel-LPC cepstrum in a large vocabulary continuous speech recognition," *Proc. ICASSP*, vol. 1, pp. 117–120, 2001.
10. Oppenheim, A.V. and Schafer, R.W., *Discrete-time signal processing.* Prentice-Hall Inc., 1989.
11. National Institute of Standards and Technology (NIST), "Rich transcription 2005 spring meeting recognition evaluation," www.nist.gov/speech/tests/rt/rt2005/spring, June 2005.
12. Linguistic Data Consortium (LDC), "Translanguage english database," LDC2002S04.

Robust Heteroscedastic Linear Discriminant Analysis and LCRC Posterior Features in Meeting Data Recognition

Martin Karafiát, František Grézl, Petr Schwarz,
Lukáš Burget, and Jan Černocký

Speech@FIT, Faculty of Information Technology, Brno University of Technology
{karafiat,grezl,schwarzp,burget,cernocky}@fit.vutbr.cz

Abstract. This paper investigates into feature extraction for meeting recognition. Three robust variants of popular HLDA transform are investigated. Influence of adding posterior features to PLP feature stream is studied. The experimental results are obtained on two data-sets: CTS (continuous telephone speech) and meeting data from NIST RT'05 evaluations. Silence-reduced HLDA and LCRC phoneme-state posterior features are found to be suitable for both recognition tasks.

1 Introduction

The AMI project[1] concentrates at processing and analysis of meetings. One of key problems is to determine what was said in the meeting; this task is accomplished by large vocabulary continuous speech recognition (LVCSR). This paper deals with the extraction of features from speech signal.

One of key problems in feature extraction is to reduce the dimensionality of feature vectors while preserving the discriminative power of features. Linear transforms such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are mostly used for this task. In recent years, Heteroscedastic Linear Discriminant Analysis (HLDA) has gained popularity in the research community [2,1] for its relaxed constraints on statistical properties of classes (Unlike LDA, HLDA does not assume same covariance matrix for all classes). To compute HLDA transformation matrix, however, more statistics need to be estimated and the reliability of such estimations becomes an issue. Section 2 discusses robust variants of HLDA.

Second part of the paper is devoted to the use of posterior-features. Posteriors generated by neural networks (NN) and converted into features are also increasingly popular in small [7] and large [6] recognition systems for their complementarity with classical PLP or MFCC coefficients. Section 3 introduces phoneme-state posterior estimator based on split temporal context [8,9] that has already proved its quality in different tasks ranging from language identification to keyword spotting.

[1] Augmented Multi-Party Interaction http://www.amiproject.org

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 275–284, 2006.

Most of the development work was done on a continuous telephone speech system (section 4). For the recognition of meetings, we used NIST RT'05 data and took advantage of the AMI-LVCSR system [5][2]. The results described in section 5 are obtained by re-scoring LVCSR lattices generated by the AMI system.

2 HLDA

HLDA allows to derive such projection that best de-correlates features associated with each particular class (maximum likelihood linear transformation for diagonal covariance modeling [2]). To perform de-correlation and dimensionality reduction, n-dimensional feature vectors are projected into first $p < n$ rows, $\mathbf{a}_{k=1...p}$, of $n \times n$ HLDA transformation matrix, $\mathbf{A}$. An efficient iterative algorithm [3,1] is used in our experiments to estimate matrix $\mathbf{A}$, where individual rows are periodically re-estimated using the following formula:

$$\hat{\mathbf{a}}_k = \mathbf{c}_k \mathbf{G}^{(k)-1} \sqrt{\frac{T}{\mathbf{c}_k \mathbf{G}^{(k)-1} \mathbf{c}_k^T}} \tag{1}$$

where $\mathbf{c}_i$ is the i^{th} row vector of co-factor matrix $C = |\mathbf{A}|\mathbf{A}^{-1}$ for current estimate of $\mathbf{A}$ and

$$\mathbf{G}^{(k)} = \begin{cases} \sum_{j=1}^{J} \frac{\gamma_j}{\mathbf{a}_k \hat{\boldsymbol{\Sigma}}^{(j)} \mathbf{a}_k^T} \hat{\boldsymbol{\Sigma}}^{(j)} & k \leq p \\ \frac{T}{\mathbf{a}_k \hat{\boldsymbol{\Sigma}} \mathbf{a}_k^T} \hat{\boldsymbol{\Sigma}} & k > p \end{cases} \tag{2}$$

where $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\Sigma}}^{(j)}$ are estimates of global covariance matrix and covariance matrix of j^{th} class, γ_j is number of training feature vectors belonging to j^{th} class and T is the total number of training feature vectors. In our experiments, the classes are defined by each Gaussian mixture component and γ_j are their occupation counts.

Well known Linear Discriminant Analysis (LDA) can be seen as special case of HLDA, where it is assumed that covariance matrices of all classes are the same. In contrast to HLDA, closed form solution exists in this case. Basis of LDA transformation are given by eigen vectors of matrix $\boldsymbol{\Sigma}_{AC} \times \boldsymbol{\Sigma}_{WC}^{-1}$, where $\boldsymbol{\Sigma}_{WC}$ is within-class covariance matrix and $\boldsymbol{\Sigma}_{AC}$ is across-class covariance matrix.

2.1 SHLDA

HLDA requires the covariance matrix to be estimated for each class. The higher number of classes is used, the fewer feature vector examples are available for each class and class covariance matrix estimates become more noisy. We have

[2] Brno University of Technology is a member of AMI-LVCSR development team, coordinated by University of Sheffield.

recently proposed [1] a technique based on combination of HLDA and LDA, where class covariance matrices are estimated more robustly, and at the same time, (at least the major) differences between covariance matrices of different classes are preserved. Smoothed HLDA (SHLDA) differs from HLDA only in the way of class covariance matrices estimation. In the case of SHLDA, estimate of class covariance matrices is given by:

$$\check{\boldsymbol{\Sigma}}_j = \alpha \hat{\boldsymbol{\Sigma}}_j + (1-\alpha)\boldsymbol{\Sigma}_{WC} \tag{3}$$

where $\check{\boldsymbol{\Sigma}}_j$ is "smoothed" estimate of covariance matrix for class j. $\hat{\boldsymbol{\Sigma}}_j$ is estimate of covariance matrix, $\boldsymbol{\Sigma}_{WC}$ is estimate of within-class covariance matrix and α is smoothing factor — a value in the range of 0 to 1. Note that for α equal to 0, SHLDA becomes LDA and for α equal to 1, SHLDA becomes HLDA.

2.2 MAP-SHLDA

SHLDA gives more robust estimation than standard HLDA but optimal smoothing factor α depends on the amount of data for each class. In extreme case, α should be set to 0 (HLDA) if infinite amount of training data is available. With decreasing amount of data, optimal α value will slide up to LDA direction.

To add more robustness into the smoothing procedure, we implemented maximum a posteriori (MAP) smoothing [4], where within-class covariance matrix $\boldsymbol{\Sigma}_{WC}$ is considered as the prior. Estimate of the class covariance matrix is then given by:

$$\check{\boldsymbol{\Sigma}}_j = \boldsymbol{\Sigma}_{WC}\frac{\tau}{\gamma_j+\tau} + \hat{\boldsymbol{\Sigma}}_j\frac{\gamma_j}{\gamma_j+\tau} \tag{4}$$

where τ is a control constant. Obviously, if insufficient data is available for current class, the prior resource $\boldsymbol{\Sigma}_{WC}$ is considered as more reliable than the class estimation $\hat{\boldsymbol{\Sigma}}_j$. In case of infinite data, only the class estimation of covariance matrix $\hat{\boldsymbol{\Sigma}}_j$ is used for further processing.

2.3 Silence Reduction in HLDA

From the point of view of transformation estimation, silence is a "bad" class as its distributions differ significantly from all speech classes. Moreover, training data (even if end-pointed) contains significant proportion of silence. Therefore, we have experimented with limiting the influence of silence.

Rather than discarding the silence frames, the occupation counts, γ_j, of silence classes, which takes part in computation of global covariance matrix, $\hat{\boldsymbol{\Sigma}}$, and in equation 2 are scaled by factor $1/SR$

$SR = \infty$ corresponds to complete elimination of silence statistics.

3 Posterior Features

Several works have shown that using posterior-features generated by NNs is advantageous for speech recognition [7,6]. We have experimented with two setups

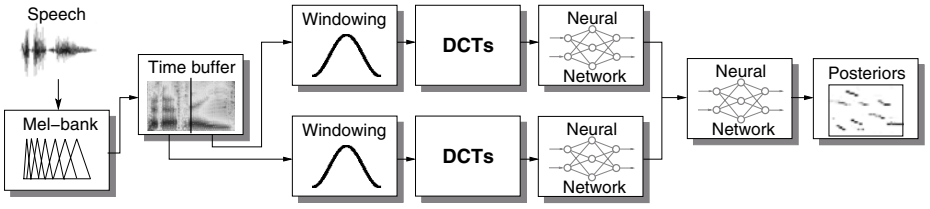


Fig. 1. Phoneme-state posterior estimator based on split left and right contexts

to generate posteriors. The first one is based on a simple estimation of phoneme posterior probabilities from a block of 9 consecutive PLP-feature vectors (FeatureNet).

The second one uses our state-of-the-art phoneme-state posterior estimator based on modeling long temporal context[9]. Details of the posterior estimator are shown in Fig. 1. Mel filter bank log energies are obtained in conventional way. Based on our previous work in phoneme recognition [8], the context of 31 frames (310 ms) around the current frame is taken. This context is split into 2 halves: Left and Right Contexts (hence the name "LCRC"). This allows for more precise modeling of the whole trajectory while limiting the size of the model (number of weights in the NN) and reducing the amount of necessary training data. For both parts, temporal evolutions of critical band log energies are processed by discrete cosine transform to de-correlate and reduce dimensionality. Two NNs are trained to produce phoneme-state posterior probabilities for both context parts. We use 3 states per phoneme which follows similar idea as states in phoneme HMM. Third NN functions as a merger and produces final set of phoneme-state posterior probabilities[3]

For both approaches, the resulting posteriors are processed by log and by a linear transform to de-correlate and reduce dimensionality (details are given in experimental sections 4 and 5).

4 CTS Experiments

Our recognition system was trained on ctstrain04 training set, a subset of the h5train03 set, defined at the Cambridge University as a training set for Conversation Telephone Speech (CTS) recognition systems [5]. It contains about 278 hours of well transcribed speech data from Switchboard I, II and Call Home English. All systems were tested on the Hub5 Eval01 test set composed of 3 subsets of 20 conversations from Switchboard I, II, and Switchboard-cellular corpora, for a total length of about 6 hours of audio data.

The baseline features are 13th order PLP cepstral coefficients, including 0th one, with first and second derivatives added. This gives a standard 39 dimension

[3] Neural nets are trained using QuickNet from ICSI and SNet – a parallel NN training software being developed in Speech@FIT.

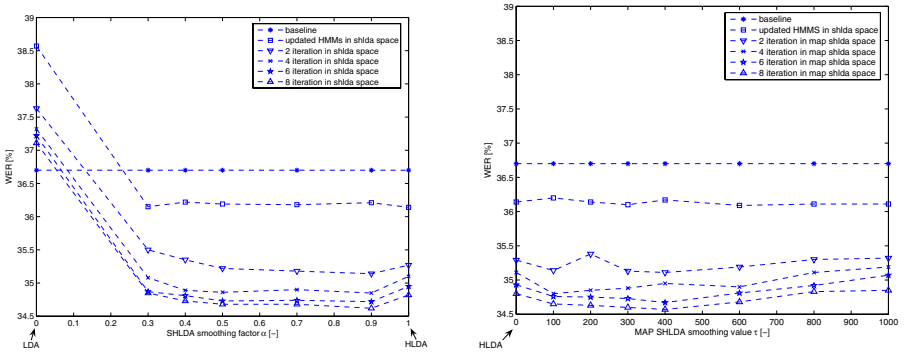


Fig. 2. Dependency of WER on the SHLDA (left) and MAP-SHLDA (right) smoothing factors

feature vector. Cepstral mean and variance normalization was applied. Baseline cross-word triphone HMM models were trained by Baum-Welch re-estimation and mixture splitting. We used a standard 3-state left-to-right phoneme setup, with 16 Gaussian mixtures per state. 7598 tied states were obtained by decision tree clustering. Each Gaussian mixture was taken as a different class for HLDA experiment. Therefore, we had $N = 16 \times 7598 = 121568$ classes.

The tri-gram language model used in decoding setup was computed by interpolation from Switchboard I, II, Call Home English and Hub4 (Broadcast news) transcriptions. The size of recognition vocabulary was 50k words.

The recognition output was generated in two passes: At first, lattice generation with baseline HMMs and bigram language model was performed. The lattices were expanded by more accurate trigram language model. The pruning process was applied to reduce them to reasonable size. In the second pass, lattices were re-scored with tested features and models.

4.1 Flavors of HLDA

We added the third derivatives into the feature stream, which gave us 52 dimensional feature vectors. **SHLDA** transform was then trained to perform the projection from 52 to 39 dimension. Smoothing factors α in Eq. 3 of 0.0 (LDA), 0.3, 0.4, 0.5, 0.7, 0.9, 1.0 (HLDA) were tested. Left panel of figure 2 shows dependency of WER on SHLDA smoothing factor α. Pure LDA failed, probably due to bad assumption of the same Gaussian distribution in all classes. The best system performance (Table 1) was obtained for smoothing factor 0.9. The relative improvement of this system is 7.9% compared to the baseline and 0.6% compared to the clean HLDA setup.

MAP-SHLDA test setup was built in same way as SHLDA system, only the smoothing procedure (Equation 3) was replaced by MAP approach (Equation 4). The average value of all class occupation counts was 820. Therefore $\tau = 820$ in MAP-SHLDA should have the same behavior as $\alpha = 0.5$ in SHLDA if all

Table 1. Comparison of HLDA systems

System	WER [%]
Baseline (no HLDA)	36.7
HLDA	34.8
SHLDA	34.6
MAP-SHLDA	34.6
SR-HLDA	34.5

classes had the same number of observations. The optimal smoothing values for SHLDA were in range 0.5—0.9 (left panel of Figure 2). Therefore, we decided to test optimal smoothing control constant τ on values 0 (HLDA), 100, 200, 300, 400, 600, 800 and 1000. The results are shown in right panel of Figure 2. The best system performance (Table 1) was obtained for $\tau = 400$. The relative improvement of this system is 8% compared to the baseline and 0.7% compared to the clean HLDA setup.

Silence reduction in HLDA (SR-HLDA) was tested with factors SR equal to 1 (no reduction), 2, 10, 100 and ∞ (removing all silence classes). For $SR = 1$, the WER is obviously 34.8%, for $SR = 2$ it drops to 34.6% and from $SR = 10 \ldots \infty$ it is constant: 34.5%.

4.2 Posterior Features

Posterior features were always used together with base PLP features. Table 2 summarizes the results.

Upper part of Figure 3 shows the way the two feature streams were combined in FeatureNet experiments. The upper branch corresponds to the previous section. To compute posterior features, 9 frames of PLP+Δ+$\Delta\Delta$ were stacked and processed by a neural net with 1262 neurons in the hidden layer (this number was chosen to have approximately 500k weights in the NN). There are 45 phoneme classes, which determines the size of the output layer. Log-posteriors are processed by KLT or HLDA and then concatenated with PLP+HLDA features to form the final 64-dimensional feature vectors.

Lower panel of Figure 3 presents the setup with LCRC-posterior features. The PLPs were derived directly with Δ, $\Delta\Delta$ and $\Delta\Delta\Delta$, and down-scaled by HLDA to 39 dimensions. The detail of LCRC-posterior feature derivation is in Fig. 1, all nets had 1500 neurons in the hidden layer. For each frame, the output of LCRC system are estimates of 135 phoneme-state[4] posterior probabilities. As the number of phoneme-state posteriors is too high to fit the statistics necessary for HLDA estimation into the memory, the output dimensionality of LCRC system is first reduced by KLT from 135 to 70. The following HLDA reduces this size to 25, and the results are concatenated with PLP+HLDA features to form again 64-dimensional feature vectors.

[4] See [9] for details on splitting each of phonemes to 3 phoneme-states.

Table 2. Performance of posterior features in the CTS system

System	WER [%]
PLP SR-HLDA	34.5
PLP SR-HLDA + PLP-posteriors KLT	33.8
PLP SR-HLDA + PLP-posteriors HLDA	33.3
PLP SR-HLDA + LCRC-posteriors HLDA	32.6

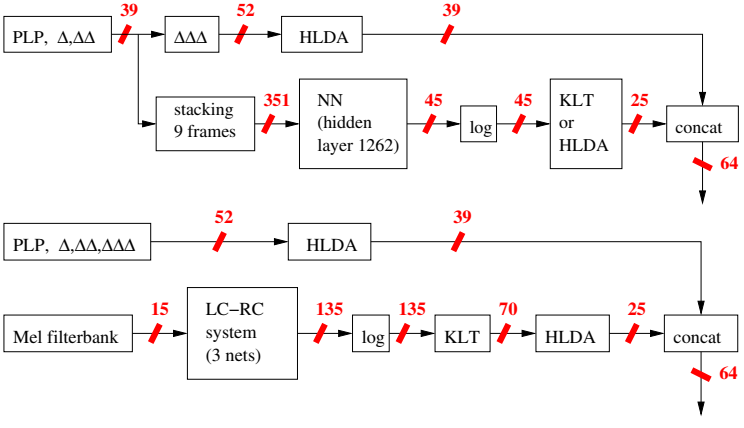


Fig. 3. Configuration of the system with PLP- (upper panel) LCRC-posteriors (lower panel)

We see, that the posterior features improve the results by almost 1% absolutely, and that there is clear preference of HLDA to KLT. With the new LCRC features, we have confirmed good results they provide in phoneme recognition [9] — with these features, the results are almost 2% better tha the PLP SR-HLDA baseline.

5 Meeting Data Experiments

Training and test sets for these experiments are the same as those used by AMI-LVCSR system for NIST RT'05 evaluations [5], therefore, we limit ourselves to the most important details. The training data consists of more than 100 hours of meeting data originating from ICSI meeting corpus (73h), NIST data (13h), ISL (10h) and AMI preliminary development set (16h).

The test data comes from NIST RT'05 and consists of two 10-minute excerpts from meetings collected by ICSI, NIST, ISL, AMI, and Virginia Polytechnic and State University (VT). NIST RT'05 included audio from headset microphones (Independent Headset Microphone, IHM) and from table-top microphones (Multiple Distant Microphones, MDM), in this work, only IHM condition was used.

Table 3. Performance of HLDA and posterior features in meeting recognition

System	WER [%]
PLP SR-HLDA	28.7
PLP SR-HLDA + LCRC-posteriors HLDA	26.0

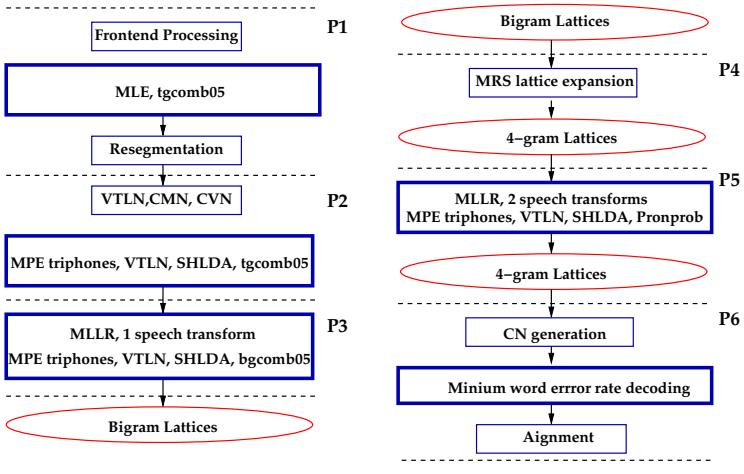


Fig. 4. Processing stages of AMI system for 2005 NIST evaluations. The figure is reprinted from [5] with permission of the author.

The system architecture is described in detail in [5]. The system operates in a total of 6 passes according to Fig. 4. The 4-gram lattices generated in step P4 were taken as input for the tests described in this section and were re-scored with different feature-extraction setups.

5.1 HLDA and Posterior Features in Meeting Recognition

Table 3 shows the results for different feature setups. The baseline for these experiments are PLP features with VTLN (VTLN was applied also prior to any posterior feature derivation) and SR-HLDA for de-correlation and dimensionality reduction (PLP $\Delta\Delta\Delta$ (52 dimensions) $\rightarrow$ 39). SR-HLDA was selected for good and "cheap" performance in the CTS system.

Posterior features were generated with the LCRC-system accordingly to Fig. 1 and lower panel of Fig. 3. The only changes from the CTS setup were the use of 23 filters in the bank instead of 15 (wide-band speech) and consistent application of VTLN prior to both PLP and Mel-filterbank computation. The sizes of hidden layers in neural nets were the same (1500 neurons), the size of input layers in left- and right-context nets increased due to increased numbers of bands. The improvement obtained by LCRC-posterior features is again quite impressive – 2.7%.

6 Conclusion

In this paper, we have investigated robust variants of HLDA and use of classical and novel posterior features in telephone speech and meeting data recognition.

In the HLDA part, 2 approaches of HLDA smoothing were tested: Smoothed HLDA (SHLDA) and MAP variant of SHLDA taking into account the amounts of data available for estimation of statistics for different classes. Both perform better than the basic HLDA. We have however found, that removing of silence class from the HLDA estimations (Silence-reduced HLDA) is equally effective and cheaper in computation. Testing SHLDA and MAP-SHLDA on the top of SR-HLDA did not bring any further improvement, therefore we stick with SR-HLDA as the most suitable transformation in our LVCSR experiments.

Two kinds of posterior features were tested – "classical" FeatureNet approach with stacked 9 frames of PLPs and novel approach using more elaborate structure to phoneme-state posterior modeling. The later scheme provided significant reduction of error rate in both CTS and meeting data experiments.

In our future work, we will investigate if the improvement obtained by LCRC-posteriors is preserved after discriminative training and speaker adaptation (MLLR) applied on the top of such features. Preliminary results are quite promising. The described features will be integrated into AMI-LVCSR system in NIST RT'06 evaluations.

Acknowledgments

This work was partially supported by EC project Augmented Multi-party Interaction (AMI), No. 506811 and Grant Agency of Czech Republic under project No. 102/05/0278. Lukáš Burget was supported by post-doctoral grant of Grant Agency of Czech Republic No. 102/06/P383. Thanks University of Sheffield for generating LVCSR lattices. We further thank Cambridge University Engineering Department making the h5train03 CTS training set available for granting the right to use Gunnar Evermann's HDecode to the University of Sheffield.

References

1. L. Burget, "Combination of speech features using smoothed heteroscedastic linear discriminant analysis," in *8th International Conference on Spoken Language Processing*, Jeju island, KR, oct 2004.
2. N. Kumar, *Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition*, Ph.D. thesis, John Hopkins University, Baltimore, 1997.
3. M.J.F. Gales., "Semi-tied covariance matrices for hidden markov models," *IEEE Trans. Speech and Audio Processing*, vol. 7, pp. 272–281, 1999.
4. J. Gauvain and C. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 291–298, Apr. 1994.

5. T. Hain et al., "The 2005 AMI system for the transcription of speech in meetings," in *Proc. Rich Transcription 2005 Spring Meeting Recognition Evaluation Workshop*, Edinburgh, July 2005.
6. Q. Zhu, A. Stolcke, B. Y. Chen and N. Morgan: "Using MLP Features in SRI's Conversational Speech Recognition System" in *Proc. Eurospeech 2005*, pp. 2141–2144, Lisabon, Portugal, 2005.
7. A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas. Qualcomm-ICSI-OGI features for ASR. In *Proc. ICSLP 2002*, Denver, Colorado, USA, 2002.
8. P. Schwarz, P. Matějka, and J. Černocký, "Towards lower error rates in phoneme recognition," in *Proc. International Conference on Text, Speech and Dialogue*, Brno, Czech Republic, Sept. 2004, pp. 465–472.
9. Petr Schwarz, Pavel Matějka and Jan Černocký: "Hierarchical structures of neural networks for phoneme recognition", accepted to ICASSP 2006, Toulouse, 2006.

Juicer: A Weighted Finite-State Transducer Speech Decoder

Darren Moore[1], John Dines[1], Mathew Magimai Doss[1], Jithendra Vepa[1], Octavian Cheng[1], and Thomas Hain[2]

[1] IDIAP Research Institute and Ecole Polytechnique Federale de Lausanne (EPFL), Martigny, Switzerland
[2] Department of Computer Science, University of Sheffield, UK

Abstract. A major component in the development of any speech recognition system is the decoder. As task complexities and, consequently, system complexities have continued to increase the decoding problem has become an increasingly significant component in the overall speech recognition system development effort, with efficient decoder design contributing to significantly improve the trade-off between decoding time and search errors. In this paper we present the "Juicer" (from trans***ducer***) large vocabulary continuous speech recognition (LVCSR) decoder based on weighted finite-State transducer (WFST). We begin with a discussion of the need for open source, state-of-the-art decoding software in LVCSR research and how this lead to the development of Juicer, followed by a brief overview of decoding techniques and major issues in decoder design. We present Juicer and its major features, emphasising its potential not only as a critical component in the development of LVCSR systems, but also as an important research tool in itself, being based around the flexible WFST paradigm. We also provide results of benchmarking tests that have been carried out to date, demonstrating that in many respects Juicer, while still in its early development, is already achieving state-of-the-art. These benchmarking tests serve to not only demonstrate the utility of Juicer in its present state, but are also being used to guide future development, hence, we conclude with a brief discussion of some of the extensions that are currently under way or being considered for Juicer.

1 Introduction

Speech recognition technology draws on a number of sources of knowledge and integrates these in the speech decoder to estimate the most likely word sequence from the given acoustical evidence. Typically these knowledge sources are represented in the form of hidden Markov models (HMM) of phonemes, a pronunciation lexicon, and N-gram language models. The means for combining these knowledge sources and efficient decoding of the acoustic input is a demanding task and a range of optimisation techniques and heuristics are employed to achieve lower computational and memory requirements with minimal sacrifice to recognition accuracy [1]. In this paper we present the "Juicer" decoding software

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 285–296, 2006.

that has been developed at IDIAP. The decoder is based on weighted finite-state transducer (WFST) theory, permitting simple decoder design through the efficient composition of a static decoding network.

We begin the paper with a short preamble, presenting our motivation for developing the Juicer decoder, followed in Section 3 by a brief overview of decoder technology and the primary design considerations, thus leading to Section 4 in which we present the Juicer system. In Section 5 we then follow-up with some preliminary benchmarking tests that have been carried out to date and in Section 6 an overview of future development directions for Juicer is given. Finally, in Section 7 we make some brief concluding remarks.

2 Why Another Speech Decoder?

Over the years many decoding software packages employing a number of different decoding strategies and sporting various capabilities have been made available to the research community and public at large, often in open source form. To name a few, there is HVite as part of HTK [4], Sphinx [10], NOWAY [7] not to forget IDIAP's own earlier effort, TODE [14]. One feature that all these decoders have in common is that they employ the acoustic, phonetic, lexical and linguistic knowledge sources in a manner that is hard-wired into the decoder architecture, thus making modifications to the decoder non-trivial. This can make the incorporation of new types of knowledge source into the decoder a significant undertaking (and possibly even infeasible for a given decoder architecture) and, as a result of this, it means that advancements to the state-of-the-art in speech recognition are often not included in the decoder and are rather used for rescoring decoder output, where their impact is likely to be more limited.

Not all decoder architectures suffer from such limitations. In recent years considerable effort has been invested in the development of more flexible decoder architectures based upon the theory of weighted finite-state transducers [13,2] in which the decoding network is compiled independently of the decoder, thus enabling a more flexible approach to the incorporation of the various speech recognition knowledge sources. This approach also has some significant drawbacks, in particular, the memory demands for the compilation of static decoding networks for LVCSR systems can quickly grow beyond the capabilities of most machines, but efforts have also been made to alleviate this problem [9,2]. While there has been significant efforts made towards developing decoder technology based upon WFST, unfortunately for the research community, to the best of our knowledge the availability of a state-of-the-art, open source decoder based upon WFST is yet to be realised.

There are many research groups around the world that are conducting significant research in LVCSR, many using their own in house recognition engine or relying on cooperation with industry for their decoder technology. In the present research environment, with many institutions and companies partnering up in European and international projects such as AMI and DARPA GALE, there is an increasing motivation for using systems and technologies that can be easily

integrated and compared. In this respect there are no 'standard' recognition system configurations and file formats, but maintaining compatibility with widely accepted technologies, using modular system and software design and using open source distribution framework can help engender collaborative speech recognition research environments.

Thus far, we have identified key motivating factors for the development of new speech decoding software. In the remaining sections we present a brief overview of speech decoding technology and, more specifically, the Juicer decoder which was developed in response to these factors.

3 LVCSR Speech Decoding

3.1 The Decoding Problem

Simply stated, the decoding problem in speech recognition is to find the most likely word sequence, $W_1^n = w_1, w_2, \ldots, w_n$, given a sequence of acoustic observation vectors, $O_1^T = o_1, o_2, \ldots, o_T$, derived from the speech signal. This can be expressed by the equation:

$$\hat{W} = \arg\max_{W_1^n} \left\{ P(W_1^n) P(O_1^T | W_1^n) \right\} \tag{1}$$

$$= \arg\max_{W_1^n} \left\{ P(W_1^n) \cdot \sum_{S_1^T} P(O_1^T, S_1^T | W_1^n) \right\} \tag{2}$$

where the sequence of words, W_1^n, is drawn from a vocabulary of size N_W, and $S_1^T = s_1, s_2, \ldots, w_T$ is any state sequence of length T.

Thus, our knowledge sources are incorporated into the decoder architecture by way of an hierarchical organisation:

- $P(W_1^n)$ comprises the language model (LM) which represents our prior **linguistic knowledge** independently of the observed acoustic information. Typically, language modelling is carried out using stochastic N-gram in which word probabilities are only dependent on the $\mathrm{N}-1$ predecessors:
- $P(O_1^T | W_1^n)$ represents our model of the lexical, phonetic, and acoustic knowledge:
  - The **lexical knowledge** comprises the known words along with their pronunciation. Multiple pronunciations may be provided with a prior probability for each pronunciation variant.
  - The **phonetic knowledge** describes the fundamental units in the pronunciation lexicon. These units are usually modelled in the context of their neighbours to account for the systematic, contextual variation that occurs in naturally spoken speech, even across word boundaries.
  - **Acoustic knowledge** is represented by way of the state emission probability density functions associated with each state of each context-dependent phoneme. In practice, various parameter tying schemes are used in emission PDF estimation.

A complete search of the solution space is practically infeasible, hence, a number of approaches are employed to solve the decoding problem in a tractable fashion. One such approach is the time-synchronous search, which under the Viterbi criterion approximates the solution to Equation 2 by only searching for the most probable state sequence:

$$\hat{W} \approx \arg\max_{W_1^n} \left\{ P(W_1^n) \cdot \max_{S_1^T} P(O_1^T, S_1^T | W_1^n) \right\} \tag{3}$$

Thus, decoding involves the time-synchronous search of a network of hypotheses, where at each time step only the best hypothesis arriving at each state is retained. To further improve efficiency only the most likely hypotheses are extended to the next time step. Such hypothesis pruning can greatly improve efficiency, but at the cost of introducing search errors. Two common pruning approaches are beam search pruning, in which hypotheses with likelihood scores falling more than a fixed amount below the highest scoring path are disregarded; and histogram pruning in which an upper threshold on the maximal number of active hypotheses at any time step is enforced, once again the hypotheses below this threshold being discarded [17]. It is worth observing that different decoding architectures tend to lend themselves to more or less effective pruning, thus making pruning an important feature in decoder design.

Another issue in decoder design is the expansion of the network which dictates the allowable hypothesis extensions, $s_t \rightarrow s_{t+1}$. This can either be carried out statically or dynamically at run-time. Static network expansion offers several advantages, in particular, the full optimisation of the decoding network and decoupling of the network expansion and decoder, but there are significant challenges in developing network expansion strategies that are not prohibitively demanding on memory resources. Converse to static network expansion, dynamic network expansion forms an integral part of the speech decoder, enabling the handling of large scale decoding tasks as the decoding network is only ever partially expanded. A consequence of this is the need to incorporate a number of sub-optimal network composition techniques that can be applied on-the-fly [17,1]. This requires the integration of network expansion and decoder, leading to a more complex and less flexible design. A compromise is also possible that involves a hybrid of these two extremes.

There is a great deal of literature available detailing the various decoding approaches and their key attributes, interested readers are referred to [1] which gives a comprehensive overview of the major decoding strategies and further references to prominent articles in the field.

3.2 WFST and Speech Decoding

While the use of static networks in speech decoding is far from being a new idea, the explicit use of weighted finite-state transducers is relatively recent. Pioneered by Mohri and others at AT&T [11], the key advantage behind the use of WFSTs for speech decoding is that it enables the integration and optimisation of all

knowledge sources within the same generic representation. This provides a more efficient framework for carrying out speech recognition and also enables greater ease for the integration of new knowledge sources in various stages of the system hierarchy. In this section we briefly describe the key features of WFST theory and its application to speech decoding.

Overview of WFST

A weighted finite-state transducer is a finite-state automaton with state transitions labelled with input and output symbols and each transition having an associated weighting. Sequences of input symbols are thus mapped to sequence of output symbols with a weighting value which is calculated over all valid paths through the transducer, where each path weight is a function of all the state-transition weights associated with that path. An example of a simple WFST is shown in Figure 1. WFST algorithms comprise a number of fundamental operations for composition and optimisation, which are briefly summarised below. Further details of the algebraic notation and algorithms for WFST can be found in [13,11].

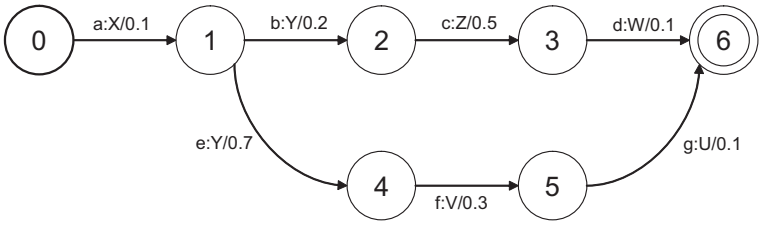


Fig. 1. An example of a simple WFST: one path through the network would have input label sequence *abcd*, output label sequence $XYZW$ and weight $f(0.1, 0.2, 0.5, 0.1)$

Composition. Composition is used to combine transducers of different levels of representation. The operation $C = A \circ B$ specifies the composition of two transducers A and B with input/output symbols x/y and y/z, respectively, into a single transducer, C, with input/output symbols x/z and weights calculated to give the same weighting to all possible input/output sequences as the original separate transducers.

Determinization. A transducer is deterministic if and only if each of it's states has at most one transition for any given input label and there are no *epsilon* input labels[3]. Determinization, denoted $\det(C)$, serves to reduce redundancy in the network thus reducing the time taken to match paths with input sequences.

Minimisation. A minimised automata, $D = \min(C)$, is equivalent to automata C and has the least number of states and the least number of transitions

[3] *epsilon* (ϵ) labels consume no input or produce no output.

among all deterministic automata equivalent to C. As the weighting of transitions tends to result in all transitions being distinct classical minimisation techniques tend to be ineffective. In order to alleviate this problem the WFST network first undergoes weight pushing in which all transitions in the transducer are reweighted to facilitate minimisation. Typically this involves a shifting of transition weights to the beginning of the network, but there is no overall effect on the total weights associated with paths through the network. It has also been demonstrated that weight pushing can be beneficial to pruning performance in ASR decoding.

Application to LVCSR

The application of WFST in LVCSR requires the representation of each of the knowledge sources as weighted finite state transducers, which subsequently undergo composition and can then be optimised using determinization and minimisation to produce a compact and efficient decoding network, as previously described. Typically, separate transducers are constructed for the N-gram language model, G, the lexicon, L, and the context dependency expansion, C. Though not currently supported in Juicer, HMM state level topology, H, and phonological information, P, can also be incorporated into the network structure:

$$N = H \circ C \circ P \circ L \circ G \tag{4}$$

In order to ensure that the entire transducer can be determinized it is necessary to undertake some additional steps:

1. In order to make the lexicon and grammar composition $L \circ G$ determinizable, the addition of an auxiliary phone symbol marking word endings in the lexicon is necessary, giving $\tilde{L}$. This auxiliary symbol must then be repeated in the transducers below lexical level (eg. $\tilde{C}$, $\tilde{P}$ and $\tilde{H}$) which at completion of determinization/minimisation undergo an erasing operation, π_ϵ, which replaces the auxiliary symbols with ϵ-labels.
2. Similarly, the context dependency transducer is generally not deterministic as there may be multiple state transitions with the same input symbol (representing the different contexts in which that symbol can occur). Building of a compact context dependency transducer can be achieved by creating the inverse of the context dependency transducer, which can be simply determinized and then inverting the resultant transducer.

Thus, the composition and optimisation of the entire static network can be expressed as follows:

$$N = \pi_\epsilon(\min(\det(\tilde{H} \circ \det(\tilde{C} \circ \det(\tilde{P} \circ \det(\tilde{L} \circ G)))))) \tag{5}$$

Further to this, additional steps may need to be taken when dealing with large vocabularies, $N_W \gtrsim 50k$ and long span language models, $N \geq 3$. Several approaches to this end have been investigated by researchers, including language model pruning, finite-state language model approximation, "on-the-fly" composition techniques and dynamic transducer composition, similar to that employed

in traditional dynamic network generation based decoders, though still employing the general WFST framework. These issues will be further touched upon in the benchmarking and future development sections in this paper.

4 An Overview of Juicer

The Juicer decoder uses a time-synchronous Viterbi search based on the token-passing algorithm with beam-search and histogram pruning, as previously described in this paper. At run time the decoder dynamically expands the model-level transducer network into a state-level network that is suitable for finding the best state-level path subject to knowledge source constraints, hence, optimisation is not yet carried out to take advantage of further state-level redundancies arising from HMM parameter sharing. The package consists of a number of command line utilities in addition to the Juicer decoder itself; more specifically, a number of tools are provided for the generation, composition and optimisation of the ASR knowledge sources (language model, pronunciation dictionary, acoustic models) into a single WFST that is input to the decoder. For the composition and optimisation of WFST resources Juicer relies on the functionality of the AT&T Finite State Machine Library [12] and/or MIT FST toolkit [6]. Figure 2 illustrates the modular organisation of the Juicer utilities.

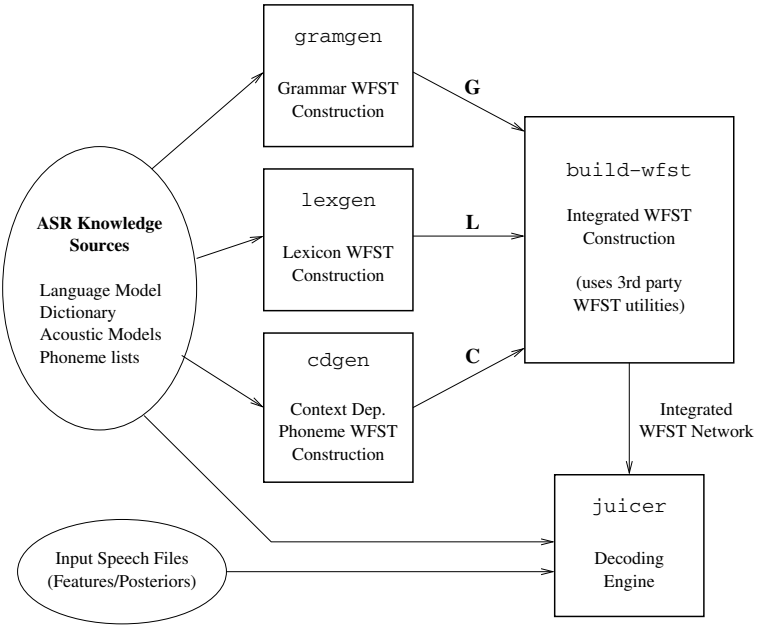


Fig. 2. High level architecture of the Juicer decoding package

The major features of Juicer and its utilities are summarised as follows; further details can be found in the user manual [15]:

- `juicer`: decoding search engine
  - Flexible WFST-based Viterbi decoder (decoding network fully independent of decoding engine implementation)
  - Beam-search (global, model-end) and histogram pruning
  - Lattice generation in AT&T FSM format
  - Word-level or model-level output with timing information
- `gramgen`: language model WFST generation
  - Simple word-loop with start/end silence
  - ARPA Naval Resource Management style word-pair grammar
  - ARPA MIT-LL text format N-gram (arbitrary N, subject to memory limitations)
- `lexgen`: dictionary WFST generation
  - Multiple pronunciations, with optional pronunciation probabilities
  - Tee models are handled via optional silence/short pause in the dictionary
- `cdgen`: acoustic model/context dependency WFST generation
  - Monophone, word-internal n-phones (tri/quin/...), cross-word triphones
  - HTK MMF file format support
  - Hybrid HMM/ANN decoding supported (using LNA-format posterior files)
- `build-wfst`: WFST composition and optimisation
  - Calls to AT&T and MIT FST routines
  - Supports optional determinization and minimisation of the final transducer (the most memory demanding step)

5 Benchmarking Experiments

Benchmarking of Juicer was carried out with two main aims; the first was to assess its performance purely from the word error rate versus pruning efficiency standpoint, and the second was to investigate its capabilities in the context of a very large vocabulary task with long span language models in which the size of the network was going to be a limiting factor.

For the first set of experiments, a system was developed using the WSJ1 continuous speech recognition corpus [16]. Three-state, cross-word triphone, decision tree state-clustered CDHMM models were trained using HTK on the "si_tr_s" set of 38,275 utterances. Models were trained from 39 dimensional MF-PLPs including delta and delta-delta features, with speaker side-based cepstral mean and variance normalisation. The pronunciation dictionary was based off that used for AMI RT05s system [5]. The standard MIT bigram and trigram backed-off language models were used with the 20k development test set "si_dt_20" from WSJ1 database, consisting of 503 utterances. Figure 3 shows the results for Juicer in comparison to HDecode, an LVCSR decoder developed at Cambridge University Engineering Department.

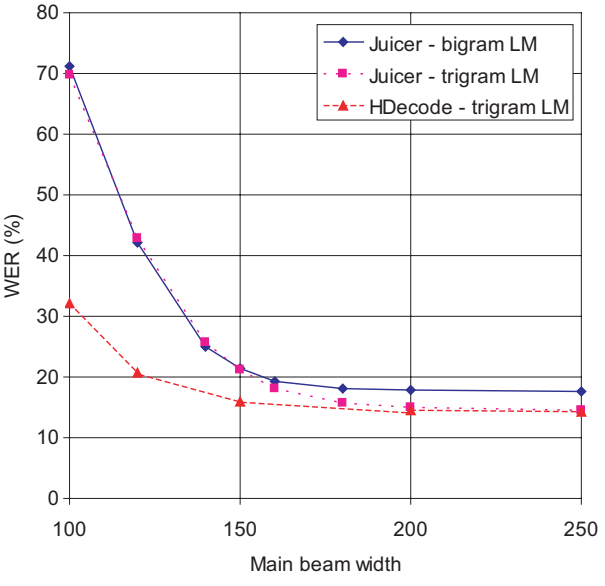


Fig. 3. Results for WSJ 20k task, showing WER versus main beam-width. Sizes of trigram transducer with Juicer (thousands of arcs): $C = 9929$, $L = 50$, $G = 15619$ and $C \circ L \circ G = 33378$.

We can see that at tight pruning settings HDecode is achieving better performance than Juicer in terms of beam-width versus word-error rate, but quickly converges within a fraction of a percent by a beam width of 200. We postulate that this originates from HDecode's use of multiple tokens per state,[4] which can be beneficial to performance by enabling the generation of more active hypothesis for the given beam width. A comparison of real-time factors would also give more insight into the differences between the two decoders, but remained outside the scope of our initial benchmarking tests as decoding experiments were conducted at different sites.

For the second set of experiments the AMI RT05s system was used one the RT05 meeting room evaluation[5]. First-pass decoding in this system uses three-state cross-word triphone models and 50k lexicon with backed-off trigram language model comprising some 29 million bigrams and 40 million trigrams, ensuring that static composition of these resources was going to be a formidable task. In order to compare the practicality of constructing a decoding network for such a system, pruned versions of the AMI language model were produced and compiled along side the full LM. Table 1 shows the outcome of the composition experiments.

We see that the size of the network grows significantly with more relaxed pruning, and in the unpruned case the final composition stage failed! In light of the size of the language model transducer this was not at all surprising and this behaviour

[4] We were unable to disable this functionality.

Table 1. Network composition experiments on RT05 system language model. DNF – did not finish. Pruned-XX – all N-grams are pruned that reduce perplexity on the training data by less than 10^{-XX} relative. The AT&T toolkit could only be used for the smallest language model.

Language Model	FSM Software	Number of arcs (thousands)					Time (hrs)
		G	L	C	$L \circ G$	$C \circ L \circ G$	
Pruned-08	AT&T + MIT	4,145	127	1,065	7,008	14,945	0:30
Pruned-09	MIT	13,692	127	1,065	23,160	50,654	1:44
Pruned-10	MIT	35,895	127	1,065	59,626	120,060	5:38
Unpruned	MIT	98,288	127	1,065	DNF	DNF	10:33

has also been reflected in the use of relatively aggressively pruned language models in some of the published literature [13]. Despite this, we were interested in evaluating the performance of Juicer against HDecode on the RT05s recognition task using a heavily pruned LM. The results are shown in Table 2. We see that despite the heavy pruning of the LM (in fact, this LM is more heavily pruned than all those shown in Table 1) the results are still resectable, with only 5% relative increase in WER. Future benchmarking experiments will look into profiling the relationship between WER and language model pruning, including the effect that this has on decoding speed, lattice generation and rescoring accuracy.

Table 2. % WER results on RT05s individual headset microphone task for HDecode (full LM) and Juicer (Pruned-07 LM). The P1 system uses ML trained models, the P2 system includes VTLN, MPE trained models, and HSLDA feature transform. Further details of the evaluation system can be found in [5].

System	TOT	Sub	Del	Ins
P1.HDecode	41.1	21.1	14.7	5.3
P1.Juicer	43.5	23.0	13.7	6.8
P2.HDecode	33.1	15.9	13.4	3.9
P2.Juicer	34.5	16.9	13.6	4.0

6 Future Development

The results of early benchmarking experiments indicate that Juicer is currently severely hampered when used for large vocabulary tasks with large, high-order N-gram language models. Hence, a priority of future development is to extend its ability with higher-order language models, however, the problem of meeting the memory requirements of such tasks through the brute force approach is seemingly unsurmountable. This is a consequence of the fact that, during composition, the size of the resultant transducer can be as large as the product of its constituents [8]. For cases of higher-order language models, we have demonstrated the composition algorithm, as well as the following optimisation procedure, can easily

fail due to lack of memory. Even if the final transducer could be successfully generated, the size may still be too large for decoding to be carried out on a conventional PC.

One possible solution to this problem is to perform on-the-fly transducer composition during decoding. Acoustical, phonetic and lexical resources may still be composed and optimised off-line, while the language model transducer is locally, dynamically composed at run time [3,18,8]. By using this approach, we can avoid composing that part of the search space which is not traversed by any hypotheses. In addition, the total size of the constituent transducers will be much smaller than the integrated transducer. This approach carries certain disadvantages in terms of introducing extra overheads during decoding and transducer optimisation operations can not be performed on the full transducer possibly leading to a sacrifice in performance with respect to pruning thresholds.

Future development of Juicer will aim to assess dynamic transducer composition along side alternative schemes, including the investigation of improved static composition techniques developed as part of the FSA toolkit, which have been demonstrated to achieve much more memory efficient composition [9] and multiple-pass decoding strategies that enable more sparse language models to be used on the first pass. Furthermore, the implementation of on-the-fly transducer composition still permits a flexible decoder architecture and need not be necessary in all applications.

7 Concluding Remarks

In this paper we have presented the Juicer speech recognition decoder developed at IDIAP. The decoder employs a statically built decoding network based upon weighted finite-state transducer theory. In benchmarking experiments we have demonstrated some of the capabilities of the decoder, in particular, we have shown that on a medium vocabulary task performance with HDecode compares favourably with moderate to wide pruning settings, while on a large vocabulary task some of the drawbacks of the current system were identified, although in spite of this, respectable WER was still able to be achieved. We have also described some of our future plans for Juicer development, more specifically, those aimed at addressing the issues raised during benchmarking. Presently, the Juicer decoder and utilities, including source code, are only available to AMI partners, but we envisage that the decoder and utilities will soon be made available to the wider research community.

Acknowledgements

The authors would like to thank all those involved in the development of the AMI ASR system, which formed the basis of some of the benchmarking evaluations that were carried out. We would also like to thank Cambridge University Engineering Department for the right to use Gunnar Evermann's HDecode at the University of Sheffield; and Adam Janin and Chuck Wooters from ICSI, who

allowed us to carry out the large network composition on one of their recently acquired 64-bit AMD Opteron Processors. This work was supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811) and the Swiss National Center of Competence in Research (NCCR) on Interactive Multi-modal Information Management (IM2).

References

1. X. Aubert. An overview of decoding techniques for large vocabulary continuous speech recognition. *Computer Speech and Language*, 16(1):89–114, January 2002.
2. D. A. Caseiro. *Finite-state methods in automatic speech recognition*. PhD thesis, Instituto Superior Téchnico, Universidade Técnica de Lisboa, December 2003.
3. H. Dolfing and I. Hetherington. Incremental language models for speech recognition using finite-state transducers. In *Proc. IEEE ASRU2001*, 2001.
4. S. Young et. al. *The HTK Book*. Cambridge University Engineering Department, December 2002. For HTK Version 3.2.1.
5. T. Hain et. al. The 2005 AMI system for the transcription of speech in meetings. In *Proc. NIST RT05 Workshop*, Edinburgh, July 2005.
6. L. Hetherington. The MIT FST toolkit. MIT Computer Science and Artificial Intelligence Laboratory: *http://people.csail.mit.edu/ilh//fst/*, May 2005.
7. M. Hochberg, S. Renals, A. Robinson, and D. Kershaw. Large vocabulary continuous speech recognition using a hybrid connectionist-HMM system. In *Proc. ICSLP*, pages 1499–1502, Yokohama, Japan, 1994.
8. T. Hori, C. Hori, and Y. Minami. Fast on-the-fly composition for weighted finite-state transducers in 1.8 million-word vocabulary continuous speech recognition. In *Proc. Interspeech (ICSLP)*, volume 1, pages 289–292, 10 2004.
9. S. Kanthak and H. Ney. FSA: An efficient and flexible C++ toolkit for finite state automata using on demand computation. In *Proc. ACL*, pages 510–517, Barcelona, Spain, July 2004.
10. K. F. Lee. *Automatic Speech Recognition – The Development of the Sphinx System*. Kluwer Academic Publishers, Norwll, Mass., 1989.
11. M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2), 1997.
12. M. Mohri, F. Pereira, and M. Riley. General-purpose finite-state machine software tools. AT&T Labs – Research: *http://www.research.att.com/sw/tools/fsm*, 1997.
13. M. Mohri, F. Pereira, and M.l Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88, January 2002.
14. D. Moore. *TODE: A Decoder for Continuous Speech Recognition*. IDIAP Research Institute, Martigny, Switzerland, 2002.
15. D. Moore. *The Juicer LVCSR decoder - user manual*. IDIAP Research Institute, Martigny, Switzerland, August 2005. for Juicer version 0.5.0.
16. D. B. Paul and J. M. Baker. The design for the wall stree journal-based CSR corpus. In *Proc. ICSLP*, 1992.
17. V. Steinbiss, B.-H. Tran, and H. Ney. Improvements in beam search. In *Proc. ICSLP*, pages 2143–2146, Yokohama, Japan, September 1994.
18. D. Willett and S. Katagiri. Recent advances in efficient decoding combining online transducer composition and smoothed language model incorporation. In *Proc. ICASSP*, volume 1, pages 713–716, 5 2002.

Speech-to-Speech Translation Services for the Olympic Games 2008

Sebastian Stüker[1], Chengqing Zong[4], Jürgen Reichert[1], Wenjie Cao[4], Muntsin Kolss[1], Guodong Xie[4], Kay Peterson[2], Peng Ding[4], Victoria Arranz[3], Jian Yu[4], and Alex Waibel[1,2]

[1] interACT, Universität Karlsruhe (TH), D-76131 Karlsruhe, Germany
stueker@ira.uka.de, juergen@ira.uka.de, kolss@ira.uka.de, waibel@ira.uka.de
[2] interACT, Carnegie Mellon University, Pittsburgh, PA 15213, USA
kay.peterson@cs.cmu.edu, waibel@cs.cmu.edu
[3] ELRA/ELDA, 75013 Paris, France
arranz@elda.org
[4] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China
cqzong@nlpr.ia.ac.cn, wjcao@nlpr.ia.ac.cn, gdxie@nlpr.ia.ac.cn, pding@nlpr.ia.ac.cn

Abstract. In 2008 the Olympics Games will be held in Beijing. For this purpose the city government of Beijing has launched the *Special Programme for Construction of Digital Olympics*. One of the objectives of the program is the use of artificial intelligence technology to overcome language barriers during the games. In order to demonstrate the contribution that speech-to-speech translation technology (SST) can make to solving this problem and in order to prove the feasibility of deploying such technology in the environment of the Olympic Games 2008 in Beijing, we have developed the *Digital Olympics Speech-to-Speech Translation System* that addresses a general touristic domain with a special focus on pre-arrival hotel reservation. The system allows for rapid development of SST prototypes, the study of different user-interfaces and the on-the-fly comparison of alternative approaches to the individual problems involved in this task.

1 Introduction

In today's world traveling the globe has become increasingly possible for a growing number of people. With the advent of affordable and fast intercontinental transportation, mostly by means of air travel, and through increasingly more transparent national borders, the number of international tourists rises steadily.

As a tourist in a foreign country one has to satisfy certain basic needs, such as shelter, food, and transportation. But when being on vacation one does not only want to fulfill these basic requirements. As a tourist one wants to interact

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 297–308, 2006.

with the people of the visited country and experience their culture. Key to this experience is the ability to communicate with the natives of the country that one visits. However, learning the language of every country one wants to travel to is clearly infeasible. In some places of the world, e.g. in Europe, English has been established as lingua franca. But even here, as in other places, English is not always spoken, especially among people with little international contact. Therefore, English as a mean of communication with the native population of an arbitrary country is often not an option. Also, language is a key component of culture. Thus, unifying the languages of the world into one common language comes with a loss in cultural diversity that we want to avoid. Here modern speech and language processing technologies can be the savior for keeping the language diversity of a globalized world.

While speech-to-speech translation for arbitrary unconstrained domains is only starting to become the topic of research activities, translation systems for limited, pre-defined domains have been developed that have reached a grade of maturity that makes them ready for field deployment in the near future.

In 2008 the Olympic Games will be held in Beijing, the capital of the People's Republic of China. It is expected that many visitors from all over the world will take this opportunity to visit China. Only very few of them will be able to speak Chinese but will want to seek deeper understanding of the Chinese culture and to come in contact with the local population. In order to make the Olympic games an attractive and enjoyable event, the city government of Beijing has launched the *Special Programme for Construction of Digital Olympics*. In the spirit of the Olympic idea one of the objectives of the program is to remove language barriers with the aid of artificial intelligence technology in order to promote friendship and mutual understanding. To achieve this goal, speech-to-speech translation technology can make a substantial contribution. In order to prove the feasibility of the deployment of speech-to-speech translation technology in the environment of the Olympic Games in 2008, we have produced the *Digital Olympics Speech-to-speech Translation System* prototype for a tourist application. The development of the prototype was a joint, international effort between *CapInfo*, the *National Laboratory of Pattern Recognition* (NLPR) at the Chinese Academy of Sciences and the *International Center for Advanced Communication Technologies* (interACT) at both Universität Karlsruhe (TH) and Carnegie Mellon University. The resulting system was successfully demonstrated at the *Beijing International High-Tech Expo* (ChiTec) 2004 in Beijing and during the *Language Technology Days* at FORUM 2004 in Barcelona.

By providing a flexible, modularized platform our system is able to demonstrate the different aspects of the technologies involved in speech-to-speech translation: automatic speech recognition, machine translation, and speech synthesis. By allowing to run different modules in parallel it is possible to compare the strengths and weaknesses of different approaches to the individual problems on the fly under real-world conditions.

2 System Overview

The *Digital Olympics Speech-to-Speech Translation System* in its current form is able to translate spontaneously spoken speech between arbitrary pairs taken from the languages Chinese, English, and Spanish. It covers a tourist expressions domain, with a strong focus on pre-arrival hotel reservation. The system integrates automatic speech recognition components, machine translation engines and speech synthesis modules and can either work as a stand-alone solution with a user interface running on a laptop or can be extended by two PDA clients that are connected via wireless network to the laptop. The two PDAs then work as the interface to the system while the translation and recognition engines are running on the laptop in a client-server setup. The system can exchange any of the six components necessary for the translating between two languages (two speech recognizers, two translation components, and two speech synthesis modules) on the fly. In that way it is possible, for example, to compare the translations for the same sentence as given by two different components at run-time.

2.1 Domain

The Digital Olympics speech-to-speech translation system demonstrates the usability of speech translation technology for the Olympic Games 2008. In order to do so, it addresses three different domains: *pre-arrival hotel reservation, basic travel expressions, basic medical needs.*

The main focus of the system is the capability to translate in the domain of pre-arrival hotel reservation. Hotel reservation is a domain in which our labs already had some experience before the beginning of the development of this system. Therefore, it was possible to develop a prototype for this scenario in a very short time. At the same time the hotel reservation scenario is close enough to the actual scenario of a foreign tourist making inquiries at a hotel desk or at a shop, so that a proof of feasibility for the pre-arrival hotel reservation scenario implies the feasibility for the other scenarios.

The domain of basic travel expressions demonstrates the capabilities of speech translation technology to act as a helper in predictable, re-occurring scenarios in which tourists need to communicate in situations typical to travelers. Currently in those situations tourists often make use of phrase books. However, PDA based translation systems can solve this issue more elegantly by providing greater flexibility in finding the correct phrases and by providing a more convenient and more natural interface. They do so by taking natural, spontaneous speech as input, giving speech output of the translation and by showing more sophisticated functionality than a phrase book. In this way, communication flows smoother than it would when utilizing an old fashioned phrasebook. This is especially true in situations where the tourist will not be able to speak the phrase book entries, as it is often the case for Chinese. As foundation for this domain we took the *Basic Traveler's Expression Corpus* (BTEC) [1].

The domain of basic medical needs can be seen as a specialized case of the travel expressions domain. Its application provides significant leverage to the

speech-to-speech translation technology due to the importance of this domain. Knowing beforehand that one will be able to overcome language difficulties in medical emergency situations will encourage people to take on a visit to a foreign country.

Fig. 1. PDA user interface

For integrating the different, often very heterogeneous, components we developed the *Active Speech* framework.

The *Active Speech* framework is an environment for building and testing multimodal interfaces. The framework allows for the easy creation of demos, prototypes, and analysis of interface issues. The basic idea of the framework is the data flow paradigm. Each component transforms received data and sends the transformed output to one, none or several receivers. The receivers themselves then again transform and resend the data. For example, speech recognizers transform audio to text, translator components transform text from one language to another, synthesis components transform text to audio etc. Special components can reconfigure the links between the components depending on what data they receive. In order to allow for distributed solutions or client-server setups, components can be located on different computers communicating over the network. Besides Windows desktops, portable devices using Windows CE are also supported.

In the *Active Speech* framework there are two kinds of components: *service components* which provide a service to other components and *visual components* which interact with the user. Visual components can be placed on the screen by drag and drop and can be connected to other components by simple mouse clicks. A system setup running in the framework can be configured and changed during run-time, in order to allow for short development cycles and interactively observe the behavior of different components in the translation task.

2.2 User Interface

The system itself has two different user interfaces, one for devices with large displays, such as laptops, and one for devices with smaller, lower resolution displays, such as PDAs. Both displays provide basically the same functionality to the user. For recording purposes it shows a waveform representation of the recorded signal. This is actually a good and intuitive display for the user to indicate the quality of the audio recording. The interface further shows the recognized sentence in the original language and the translation into the target language. In between is the phrasebook that automatically picks sentences that are close to the recognized one. In addition to the PDA user interface, the interface of the laptop allows for control of the loaded components, the selection of the components to be used in the translation process, and the manual repetition of translations and synthesis, e.g. after changing a component or correcting an error, e.g. via keyboard. Figure 2.1 shows the user interface for the PDA.

3 Speech Recognition Component

For the speech recognition components we developed large vocabulary spontaneous speech recognition systems in the three languages Chinese, English, and Spanish. The Chinese speech recognition component was developed by NLPR, the Spanish and English systems by Universität Karlsruhe (TH). Different linguistic resources for fitting the recognizers to the new domain were available, among them BTEC and data we collected at our labs, such as prototype dialogs for hotel reservation, in domain word lists, and protocols of previous demos.

3.1 Chinese Recognition System

The NLPR Chinese recognition system extracts features from overlapping 24ms long frames of audio data at a rate of 100 frames per second. From these frames 12 Mel-warped cepstral coefficients are extracted. Together with the normalized energy plus 1 dimension normalized pitch, and their first and second derivatives, they compose the final 42-dimensional feature vector.

Decision tree based gender dependent class triphones based on 62 phones (including silence as a separated phone) are trained on 800 hours of speech. Each state is modeled by 16 Gaussian components. To incorporate tone information, the question set is specially designed to take the Mandarin tonal information into consideration. The obtained acoustic model typically includes tonal information for both, the right and left context phones, and the base phone itself, and can provide detailed information for a tonal language, such as Mandarin.

The main language model used by the system is a word based N-Gram model using Katz backoff, and is estimated for a 50K word set vocabulary using hundreds of millions of words of training texts. We have also investigated the use of class-based language models using automatically derived word classes to smooth the word model probabilities in the system. The final LM used in the system is an interpolation of a 4-Gram and a class based 3-Gram model. Our LVCSR system uses

a time-synchronous decoder that can either operate in a single pass or can be used to generate or rescore word lattices. The decoder can operate with cross-word triphone models and direct incorporation of trigram language model. Afterwards the output lattice can be rescored with more advanced language models. To fit the proposed tonal AM, the search engine was updated to consider the tone information during path propagation, merge and pruning which make our decoder quinphone like.

3.2 English Recognition System

The English and Spanish recognition systems were developed with the help of the Janus Recognition Toolkit JRTk featuring the IBIS one-pass decoder[2]. The English recognition system is derived from the *ISL Meeting Transcription System*[3]. For the translation system the Meeting System was reclustered into a fully-continuous system with 6000 models, each being a Gaussian Mixture models with 32 Gaussians. The system uses an MFCC based front-end with per utterance Cepstral Mean Subtraction and Cepstral Variance Normalization. The vocabulary of the resulting system contains 2500 entries. As language model we use a 3-Gram language model with Kneser-Ney backoff trained on 2.1M word corpus with the help of the *SRI Language Model Toolkit*. In the language model we make use of manually defined semantic classes, such as hotel names, person names, local points of interest, etc. In that way it is possible to port the recognizer to new scenarios, e.g. a new city, without the need of retraining the language model.

During decoding we employ incremental vocal tract length normalization and feature space constrained MLLR on a sliding window in order to be able to adapt to changing speakers. We evaluated the system on a collection of 13 dialogs in which hotel reservation scenarios were reenacted in a spontaneous manner. The scenarios were collected with the help of 17 native speakers of American English at Carnegie Mellon University. The database contains approximately 23 minutes of speech in 319 turns. On this set the recognizer achieved a word error rate of 18.1%. On a laptop with a Pentium M 1.7GHz the system runs in realtime.

3.3 Spanish Recognition System

The Spanish system was also trained with the help of the JRTk and derived from a Global Phone speech recognition system for South American Spanish [4]. The phoneme set was modified to better fit Castilian Spanish and then trained on ca. 14h of Castilian speech. The acoustic model consists of 2000 triphones with roughly 51k Gaussians. The recognition system has a vocabulary of 24k. A 3gram language model with the same semantic classes as for the English system was trained on roughly 255k words. The training corpus includes the Spanish portion of the BTEC and several in-house, in domain text collections.

For decoding the IBIS decoder was also used, and the same incremental adaptation scheme as for the English recognizer was applied. The sytem was tested on hotel reservation dialogs collected at Universität Karlsruhe (TH) with the help of 18 native speakers. The database contains 2973 turns, giving about 170 minutes

of speech. The system yielded a word error rate of 16.3% on these dialogs. On a laptop with a Pentium M 1.7GHz the system runs in about realtime.

4 Speech Translation

For the *Digital Olympics SST System* we incorporated two different approaches to machine translation. One rule-based approach based on an interlingua described in section 5 and one statistical machine translation approach described in detail in section 6. Also, in addition to the machine translation approaches we have incorporated a phrase book constructed out of the BTEC.

4.1 Phrasebook

The phrase book contains 162320 phrases taken out of the BTEC, which were categorized into 13 classes by NLPR, and a dictionary with about 100000 translation pairs. In order to restrict the result set to a query, the user can select categories to filter the output phrases. The phrase book can be used in multiple modes. It can be used like a normal electronic dictionary or the user can search for phrases with the help of key words. The similarity mode returns related phrases to a given phrase, with a maximum word distance. This often allows to correct errors from the speech recognition, if a similar or the same phrase to the spoken one is in the database.

4.2 Translation Modes

Our speech-to-speech translation system can run in two different modes. In the first mode one can pre-select one of the provided translation components as the default for translating the speech input, e.g. the IF based translation component. The second mode is a cascade of different stages that are triggered one after the other as necessary. In the first stage a look-up in the internal phrasebook is made, to see whether a translation of the input sentence is stored there. If so, the stored translation is taken. If not the input sentence is forwarded to the IF based translation component. If the IF translation component however fails to parse the input sentence, e.g. because it is out-of-domain, the input is forwarded to the last fallback stage, the statistical machine translation component which is guaranteed to produce a translation.

5 Interlingua Based Translation

The interlingua-based engine of the MT system utilizes the Interchange Format (IF) that was originally developed for CSTAR II and later on expanded and modified for the LingWear [5] and NESPOLE! [6] systems. This interlingua was designed to cover spontaneous task-oriented dialogs in specific, limited domains [7]. It captures speaker intention rather than literal meaning, abstracting away from

syntax and language-specific idiosyncrasies. Utterances to be represented in IF are segmented into semantic dialog units (SDUs).

The assumption underlying such a domain-action based interlingua for MT purposes is that utterances relevant to a particular domain can be classified into a limited number of domain actions [8]. For each language, analysis components from input language to interlingua and generation components from interlingua to output are written to cover those domain actions reliably. Consequently, an MT engine based on this interlingua can be expected to perform best on translating utterances highly relevant to the domain it was designed for, and exhibit more limited or no translation capabilities for out-of-domain utterances. The original domain covered for CSTAR II was pre-arrival hotel reservation; for NESPOLE!, this was expanded to cover more general tourism-related inquiries and later on medical patient-doctor conversations.

5.1 Interlingua-Based Analysis and Generation

Interlingua-based MT allows for the use of different analysis and generation components for different languages that can all be integrated into the same MT engine. For the system at hand, the English and Spanish analysis and generation use the same parsing and generation mechanisms, while Chinese uses different ones and is therefore described in a separate section.

For both the analysis and generation side, the English grammars for the Digital Olympics system were expanded and adapted from pre-existing grammars. The expansion and adaptation was also done at Carnegie Mellon. Some of the structure of the English grammars was taken as a seed for the Spanish grammars, which were otherwise written from scratch specifically for the FAME [9] and Digital Olympics projects. The Spanish grammars were developed at Universitat Politècnica de Catalunya. The data used for grammar development were taken from the IF-annotated C-STAR II database, which was translated into Spanish.

5.2 English and Spanish Interlingua-Based Analysis

The SOUP parser [10], a stochastic, top-down, chart-based parser specifically developed for parsing spontaneous speech in real time, is used for English and Spanish analysis into IF. In a pre-processing step, the speech recognition output is standardized to optimize parsing performance; for example, filler words and hesitations may be removed before parsing. The SOUP parser calls context-free grammar rules from the grammar for the respective language that contain top-level domain action and lower level rules. SOUP segments utterances into SDUs at parse time. The SOUP output is then mapped into standard IF format in a subsequent step.

5.3 English and Spanish Interlingua-Based Generation

The GenKit generator [11], a powerful pseudo-unification-based generation tool, generates English and Spanish output from IF. GenKit uses language-specific hybrid syntactic/semantic grammars in combination with generation lexica to generate natural language text output from an IF-based feature structure. Highly

domain-relevant, frequent domain actions may be generated via rules specific to them to guarantee reliable and highly fluent output. On the other hand, more general rules serve as fall-back rules to enable generation from as wide a variety of domain actions as possible. GenKit grammars use syntactic, lexical, and morphological knowledge. Generation of the correct morphological form is achieved via inflectional grammar rules that draw on additional information stored in the lexical entries. In the more complex case of verb morphology in Spanish, the correct form is retrieved from an additional morphological form look-up table. Also specific to Spanish, several linguistic phenomena were tackled that had not been implemented in GenKit-based grammars before. At the end, the GenKit output is sent through a post-processing stage to ensure that clean text is produced.

5.4 Evaluation of English-Spanish IF Based Translation

In [12] the English-Spanish IF based translation was evaluated by human judgment of the fidelity and naturalness of the translations on ten hotel reservation dialogs with an English speaking client and a Spanish speaking agent. When dealing with perfect transcriptions, instead of error prone automatic transcriptions obtained from the described speech recognizers, 91.7% of the translations from Spanish to English were good in content, only 1.2% were considered bad, the rest O.K. From English to Spanish 82.4% were good in content, only 1.6% bad, the rest O.K. When using speech recognition output as input to the translation, for Spanish to English still 96.4% of the translations were at least O.K.; but only 62.4% for English to Spanish, showing the higher complexity of that direction for speech recognition as well as for translation.

5.5 Parsing and Generation of Chinese Sentences

The parsing of the Chinese sentences into IF uses a robust hybrid parsing scheme [13] that combines Hidden Markov Model (HMM) based approaches with rule based ones for extracting semantic information and mapping parsed result into IF. The parsing takes place in three distinct steps: *chunk identification, chunk and HMM based analysis, chunk interpretation.*

A chunk is defined under the restrictions of semantic level and syntactic level, which is a head-word cored semantic unit that has relatively independent phrase structure and relatively complete semantic composition. In our paraphrase system, we classify all words in the domain of travel information into 324 semantic types. Similar to the syntactic parsing with context free grammar (CFG) and Chart parsing algorithm, we have developed a grammar for semantic chunk recognition, in which all rules are described by semantic type markers. From the resulting chunk sequence the HMM based analysis picks up the skeleton of the IF. Hereby The chunk sequence is interpreted as the observation of a HMM while the IF is corresponds to the internal states of the HMM. The parameters of the HMM were trained on a large set of tagged spoken Chinese sentences. The IF skeleton for an input chunk sequence is then found by calculating the most likely HMM state sequence with the help of the Viterbi algorithm. Finally our chunk interpreter fills the

slots of the IF skeleton by using the internal structure and semantic information of the chunks acquired during the chunk identification according to the corresponding parsing rule in the rule base.

For the Chinese generation part, we implemented a feature-based generation approach in our spoken Chinese paraphraser for generating the Chinese translation from the generated IF [14].

The spoken Chinese generator consists of two functional modules, the micro-planner and the surface generator. The micro-planner generates the functional structure of the resulting sentence.

The surface generation is then the final stage of the sentence generation. The final sentence is generated based on the micro-planning results and a system functional grammar for the generation language. Our surface generator employs a top-down and depth-first unification algorithm. After that, in the sentence linearization, the order of components in the generating sentence is arranged according to the unification order of the sentence and phrase. In a post-processing step modifier words expressing the tense and voice are being added.

6 Statistical Machine Translation

Statistical machine translation is based on the noisy channel approach. The ISL system uses as primary building blocks phrase-to-phrase translations extracted from bilingual data by optimizing a constrained word-to-word alignment for an entire sentence pair [15]. Phrase translation candidates are scored by a modified IBM1 alignment model. For words inside the source phrase the summation of the lexicon probabilities is restricted to the probabilities for words inside the target phrase candidate, and for words outside of the source phrase it is restricted to the probabilities for the words outside. The alignment probabilities from both alignment directions are interpolated. Single source words are treated as phrases of length 1. Most phrase pairs are seen only a few times, even in very large corpora. Therefore, probabilities based on occurrence counts have little discriminative power. Phrase translation probabilities are calculated based on a statistical lexicon, i.e. on the word translation probabilities. The language model used in the decoder is a standard 3-gram language model. We use the SRI language model toolkit to build language models of different sizes, using the target side of the bilingual data only or using additional monolingual data. The decoding process works in two stages: First, the word-to-word and phrase-to-phrase translations are used to generate a translation lattice. Second, a first-best search is performed on this lattice, using the language model probabilities in addition to the translation model probabilities to find the overall best translation.

For training data we compiled a trilingual corpus of spoken utterances covering the target domains of general tourism, hotel reservation, medical assistance, and specific tourist assistance for the Olympic Games in Beijing. The main components of this corpus were the BTEC, as well as in-house collected dialogs for medical assistance, and hotel reservation dialogs collected at Carnegie Mellon University and translated from English into Chinese and Spanish. In addition, a manual list

of about 4600 named entities specific for the city of Beijing was added, consisting of the names of bus and metro stations, tourist attractions and sites, hotel names, and person names. Overall, the corpus currently has a size of about 190K utterance and named entity tuples. As a preprocessing step, the Chinese part of the corpus was segmented into words using a segmenter derived from the LDC segmenter. The final system also uses a small number of translation rules for number and date expressions.

7 Speech Synthesis

Concatenative speech synthesis technology has been employed in the Chinese TTS system. The system contains three main parts: text analysis model, prosody generation model, and unit selection model. The text analysis applies some preprocessing such as text normalization, parsing, and text-to-pinyin conversion. In the prosody generation model, special attention has been paid to tones during the prosody information prediction process. Depending on the prosody information and segmental information, the synthesizer selects units from a real waveform corpus, and smoothens the pitch contour before outputting the final speech.

The English and Spanish speech synthesis were provided by Cepstral LLC, Pittsburgh, PA, USA. Cepstral provides state-of-the-art unit selection text-to-speech synthesis and voices that are small and fast enough to run on handheld devices or distribute over the network. Cepstral voices support SSML, VoiceXML tags, and Microsoft(R) SAPI.

8 Conclusion

In this paper we have introduced our *Digital Olympics Speech-to-Speech Translation System*. It incorporates modules from our different labs for the technologies necessary for SST: Speech recognition, machine translation, and speech synthesis. Its successful implementation and demonstration on various occasions gives proof of the feasibility of deploying speech-to-speech translation technology in the environment of the Olympics Games 2008 in Beijing. Automatic speech-to-speech translation for tourist domains, such as general tourist needs, hotel inquiries or medical needs can provide significant leverage for the distribution of this technology due to the practical significance of the domains in real-life. Morevoer, the introduced Active Speech framework allows for fast development of prototypes, studies of user interfaces, and on-the-fly comparison of different approaches to the technologies necessary for translation systems of this kind.

Acknowledgements

The authors would like to thank Raquel Tato and Marta Tolos for their help in the development of the Spanish recognition system, Dorcas Alexander for her contribution to the development of the IF components, and Victoria MacLaren for her

help in the data collection. Special thanks go also to Elisabet Comelles for her help in the development of the IF components and the carrying out of the corresponding evaluation.

References

1. Kikui, G., Sumita, E., Takezawa, T., Yamamoto, S.: Creating corpora for speech-to-speech translation. In: EUROSPEECH. (2003)
2. Soltau, H., Metze, F., Fügen, C., Waibel, A.: A one pass-decoder based on polymorphic linguistic context assignment. In: ASRU. (2001)
3. Metze, F., Jin, Q., F ügen, C., Laskowski, K., Pan, Y., Schultz, T.: Issues in meeting transcription - the ISL meeting transcription system. In: ICSLP. (2004)
4. Schultz, T., Waibel, A.: Language independent and language adaptive acoustic modeling for speech recognition. Speech Communication **35** (2001)
5. Fügen, C., Westphal, M., Schneider, M., Schultz, T., Waibel, A.: LingWear: A mobile tourist information system. In: HLT. (2000)
6. Metze, F., McDonough, J., Soltau, H., Langley, C., Lavie, A., Levin, L., Schultz, T., Waible, A., Cattoni, R., Lazzari, G., Mana, N., Pianesi, F., Pianta, E.: The NESPOLE! speech-to-speech translation system. In: HLT. (2002)
7. Levin, L., Gates, D., Lavie, A., Waibel, A.: An interlingua based on domain actions for machine translation of task-oriented dialogues. In: ICSLP. (1998)
8. Levin, L., Langley, C., Lavie, A., Gates, D., Wallace, D., Peterson, K.: Domain specific speech acts for spoken language translation. In: 4th SIGdial Workshop on Discourse and Dialogue. (2004)
9. : (http://isl.ira.uka.de/fame/)
10. Gavalda, M.: Soup: A parser for real-world spontaneous speech. In: 6th IWPT. (2000)
11. Tomita, M., Nyberg, E.: Generation kit and transformation kit, version 3.2, user's manual. In: Technical Report CMU-CMT-88-MEMO, Carnegie Mellon University, Pittsburgh, PA, USA (1988)
12. Arranz, V., Comelles, E., Farwell, D., Nadeu, C., Padrell, J., Febrer, A., Alexander, D., Peterson, K.: A speech-to-speech translation system for catalan, spanish and english. In: AMTA. (2004)
13. Xie, G., Zong, C., Xu, B.: Chinese spoken language analyzing based on combination of statistical and rule method. In: ICSLP. (2002)
14. Cao, W.: Approach to target language generation in spoken language translation (in Chinese). Master's thesis, Institute of Automation, Chinese Academy of Sciences (2004)
15. Vogel, S., Hewavitharana, S., Kolss, M., Waibel, A.: The ISL statistical machine translation system for spoken language translation. In: IWSLT. (2004)

The Rich Transcription 2006 Spring Meeting Recognition Evaluation

Jonathan G. Fiscus[1], Jerome Ajot[1], Martial Michel[1,2], and John S. Garofolo[1]

[1] National Institute of Standards and Technology, 100 Bureau Drive Stop 8940, Gaithersburg, MD 20899
[2] Systems Plus, Inc., One Research Court – Suite 360, Rockville, MD 20850
{jfiscus,ajot,martial.michel,jgarofolo}@nist.gov

Abstract. We present the design and results of the Spring 2006 (RT-06S) Rich Transcription Meeting Recognition Evaluation; the fourth in a series of community-wide evaluations of language technologies in the meeting domain. For 2006, we supported three evaluation tasks in two meeting sub-domains: the Speech-To-Text (STT) transcription task, and the "Who Spoke When" and "Speech Activity Detection" diarization tasks. The meetings were from the Conference Meeting, and Lecture Meeting sub-domains. The lowest STT word error rate, with up to four simultaneous speakers, in the multiple distant microphone condition was 46.3% for the conference sub-domain, and 53.4% for the lecture sub-domain. For the "Who Spoke When" task, the lowest diarization error rates for all speech were 35.8% and 24.0% for the conference and lecture sub-domains respectively. For the "Speech Activity Detection" task, the lowest diarization error rates were 4.3% and 8.0% for the conference and lecture sub-domains respectively.

1 Motivation

The National Institute of Standards and Technology (NIST) has been working with the speech recognition community since the mid 1980s to improve the state-of-the-art in speech processing technologies. To facilitate progress, NIST has worked with the community to make training/development data collections available for several speech domains. NIST collaborated with the research community to define performance metrics and create evaluation tools for technology developers to perform hill-climbing experiments and measure their progress. NIST also coordinates periodic community-wide benchmark tests and technology workshops to inform the research community and Government sponsors of progress, and to promote technical exchange. The test suites used in these benchmark tests are generally made available to the community as development tools after the formal evaluations.

NIST evaluations have demonstrated great progress in the state-of-the-art in speech-to-text (STT) transcription systems[1]. STT systems in the late 80s focused on read speech from artificially-constrained domains. As the technology improved, the NIST evaluations focused the research community on increasingly difficult challenges with regard to speech modality, speaker population, recording characteristics, language, vocabulary, etc.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 309 – 322, 2006.

The meeting domain presents several new challenges to the technology. These include varied fora, an infinite number of topics, spontaneous highly interactive and overlapping speech, varied recording environments, varied/multiple microphones, multi-modal inputs, participant movement, and far field speech effects such as ambient noise and reverberation. In order to properly study these challenges, laboratory-quality experiment controls must be available to enable systematic research. The meeting domain provides a unique environment to collect naturally-occurring spoken interactions under controlled sensor conditions.

The Rich Transcription Spring 2006 (RT-06S) Meeting Recognition evaluation is part of the NIST Rich Transcription (RT) series of language technology evaluations [1] [2] [7]. These evaluations have moved the technology focus from a strictly word-centric approach to an integrated approach where the focus is on creating richly annotated transcriptions of speech, of which words are only one component. The goal of the RT series is to create technologies to generate transcriptions of speech which are fluent and informative and which are readable by humans and usable in downstream processing by machines. To accomplish this, lexical symbols must be augmented with important informative non-orthographic metadata. These resulting metadata enriched transcripts are referred to as "rich transcriptions". These metadata can take many forms (e.g., which speakers spoke which words, topic changes, syntactic boundaries, named entities, speaker location, etc.)

The RT-06S evaluation is the result of a multi-site/multi-national collaboration. In addition to NIST, the organizers and contributors included: Athens Information Technology (AIT), the Augmented Multiparty Interaction (AMI) program, the Computers in the Human Interaction Loop (CHIL) program, Carnegie Mellon University (CMU), Evaluations and Language resources Distribution Agency (ELDA), IBM, International Computer Science Institute and SRI International (ICSI/SRI), Institut National de Recherche en Informatique et Automatique (INRIA), The Center for Scientific and Technological Research (ITC-irst), Karlsruhe University (UKA), the Linguistic Data Consortium (LDC), Laboratoire Informatique d'Avignon (LIA), Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI), Universitat Politècnica de Catalunya (UPC), and Virginia Tech (VT).

Two tests were built for the evaluation with different types of meeting data: the Conference Meeting meeting sub-domain and Lecture Meeting meeting sub-domain test sets. The two test sets fostered collaboration between the many research programs by providing backward compatible test sets (previous evaluations used Conference Meeting data) and sharing data across programmatic boundaries while accommodating individual programmatic needs.

1.2 Rich Transcription Relation to Multi-modal Technology Evaluations

Beginning in the early 2000's, a number of independent meeting-domain focused research and evaluation efforts were started: the European Union (EU) Computers in the Human Interaction Loop (CHIL), the EU Augmented Multiparty Interaction (AMI) program, the US Video Analysis and Content Extraction (VACE) program, and the NIST Rich Transcription Evaluation series which shared many aspects of uni-modal and multi-modal research. Since the recognition of human-human communications in

meetings is multi-modal by nature, NIST decided to expand the evaluations it supports in this area to facilitate the development of a multi-modal research community.

NIST decided to take several steps to create a collaborative international evaluation effort that would share knowledge and resources across research programs both in the US and abroad, leverage efforts, standardize data, metrics, and interchange formats across efforts, and help increase the critical mass of research in multi-modal meeting understanding technologies. Advisory committees were established to develop plans for upcoming evaluations and workshops that selected cross-program evaluation tasks to support. As a result, the RT evaluation became a program-independent evaluation forum for language technologies with a focus on the meeting domain and the extraction of language content from both audio and video source channels. The second result was to create the Classification of Events, Activities, and Relationships (CLEAR) evaluation and workshop which focuses on spatial analysis problems.

During 2006, RT remained co-located with the 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI-06) and the CLEAR workshop occurred earlier as a separate event. For 2007, both the RT and CLEAR workshops will be co-located so that initial discussions regarding fusion technologies can be discussed and future evaluations can be planned accordingly.

2 Rich Transcription Spring 2005 Meeting Recognition Evaluation

The RT-06S evaluation was similar to the RT-05S evaluation. Two major changes were made to the evaluation: first, the Source Localization evaluation task was moved to the Classification of Events, Activities, and Relationships (CLEAR) [8] evaluation and second, overlapping speech was evaluated in both the STT and Diarization "Who Spoke When" (SPKR) tasks instead of restricting the scoring to only non-overlapping speech.

All participating teams were required to submit a single primary system on the required task-specific evaluation condition. The primary systems are expected, by the developers, to be their best performing systems. NIST's analysis focuses on these primary systems.

The Rich Transcription Spring 2006 Evaluation plan [3] describes in detail the evaluation tasks, data sources, microphone conditions, system input and output formats, and evaluation metrics employed in the evaluation. This section summarizes the evaluation plan by discussing the meeting sub-domains in the test set, the audio input conditions, the evaluation task definitions, and the evaluation corpora details.

2.1 Meeting Sub-domains: Conference Room vs. Lecture Room

The meeting domain is highly variable along several dimensions. In the broad sense, any interaction between 2 more people may be considered to be a meeting. As such, meetings can range from brief informal exchanges to extremely formal proceedings with many participants following specific rules of order. It is well known that the type, number, and placement of sensors have a significant impact on the performance of recognition tasks. The variability is so large that it would be impossible to build ei-

ther a training or testing corpus that encompasses all of these factors. To make the problem tractable, the RT evaluations have attempted to focus efforts on two specific sub-domains: small conference room meetings (also occasionally referred to as "board room" meetings) and classroom-style lectures in a small meeting room setting. The two sub-domains are used to differentiate between two very different participant interaction modes as well as two different sensor setups. The RT-06S evaluation includes a separate test set for each of these two sub-domains, labeled "*confmtg*" and "*lectmtg*."

In addition to differences in room and sensor configuration, the primary difference between the two sub-domains is in the group dynamics of the meetings. The RT conference meetings are primarily goal-oriented, decision-making exercises and can vary from moderated meetings to group consensus building meetings. As such, these meetings are highly-interactive and multiple participants contribute to the information flow and decisions made. In contrast, lecture meetings are educational events where a single lecturer briefs an the audience on a particular topic. While the audience occasionally participates in question and answer periods, it rarely controls the direction of the interchange or the outcome.

Section 2.4 describes the corpora used for both the *lectmtg* and *confmtg* domains in the RT-06S evaluation.

2.2 Microphone Conditions

Seven input conditions were supported for RT-06S. They were:

- Multiple distant microphones (MDM): This evaluation condition includes the audio from at least 3 omni-directional microphones placed (generally on a table) between the meeting participants.
- Single distant microphone (SDM): This evaluation condition includes the audio of a single, centrally located omni-directional microphone for each meeting. This microphone channel is selected from the microphones used for the MDM condition. Based on metadata provided with the recordings, it is selected so as to be the most centrally-located omni-directional microphone.
- Individual head microphone (IHM): This evaluation condition includes the audio recordings collected from a head mounted microphone positioned very closely to each participant's mouth. The microphones are typically cardioid or super cardioid microphones and therefore the best quality signal for each speaker. Since the IHM condition is a contrastive condition, systems can also use any of the microphones used for the MDM condition.
- Multiple Mark III microphone Arrays (MM3A): This evaluation condition includes audio from all the collected Mark III microphone arrays. A Mark III microphone arrays is a 64-channel, linear topology, digital microphone array[11]. The *lectmtg* dataset contains the data from each channel of one or tow Mark-III microphone array per meeting.
- Multiple Source Localization microphone arrays (MSLA): This evaluation condition includes the audio from all the CHIL source localization arrays (SLA). An SLA is a 4-element digital microphone array arranged in an upside down 'T'

topology. The lecture room meeting recordings include four or six SLAs, one mounted on each wall of the room.

- All Distant Microphones (ADM): This evaluation conditions permits the use of all distant microphones for each meeting. This condition differs from the MDM condition in that the microphones are not restricted to the centrally located microphones and the Mark III arrays and Source Localization arrays can be used. This condition was new for RT-06S.
- Multiple BeamFormed signals (MBF): This evaluation condition permits the use of the just the blind source separation-derived signals from the Mark-III arrays. This condition was new for RT-06S.

The troika of MDM, SDM, and IHM audio input conditions makes a very powerful set of experimental controls for black box evaluations. The MDM condition provides a venue for the demonstration of multi-microphone input processing techniques. It lends itself to experimenting with beamforming and noise abatement techniques to address room acoustic issues. The SDM input condition provides a control condition for testing the effectiveness of multi-microphone techniques. The IHM condition provides two important contrasts: first, it effectively eliminates the effects of room acoustics, background noise, and most simultaneous speech, and second it is most similar to the Conversational Telephone Speech (CTS) domain [1] and may be compared to results in comparable CTS evaluations.

2.3 Evaluation Tasks

Three evaluation tasks were supported for the RT-05S evaluation: a speech-to-text transcription task and two diarization tasks: "Who Spoke When" and "Speech Activity Detection". The following is a brief description of each of the evaluation tasks:

Speech-To-Text (STT) Transcription: STT systems are required to output a transcript of the words spoken by the meeting participants along with the start and end times for each recognized word. For this task, no speaker designation is required. Therefore, the speech from all participants is to be transcribed as a single word output stream.

Systems were evaluated using the Word Error Rate (WER) metric. WER is defined to be the sum of system transcription errors, (word substitutions, deletions, and insertions) divided by the number of reference words and expressed as a percentage. It is an error metric, so lowers scores indicate better performance. The score for perfect performance is zero. Since insertion errors are counted, it is possible for WER scores to exceed one hundred percent.

WER is calculated by first harmonizing the system and reference transcript through a series of normalization steps. Then the system and reference words are aligned using a Dynamic Programming solution. Once the alignment mapping between the system and reference words is determined, the mapped words are compared to classify them as either correct matches, inserted system words, deleted reference words, or substituted system words. The errors are counted and statistics are generated.

The MDM audio input condition was the primary evaluation condition for the STT task for both meeting sub-domains. The *confmtg* data set also supported the SDM and

IHM conditions. The *lectmtg* data supported the SDM, IHM, MSLA, MM3A, and MBF conditions. Participants could submit systems for the *confmtg* domain, the *lectmtg* domain, or both sub-domains.

For the RT-06S evaluation, the distant microphone systems were evaluated on speech including up to 4 simultaneous speakers. Previous evaluations ignored overlapping speech for these conditions. To compute these scores, the ASCLITE [9] module of the NIST Scoring Toolkit (SCTK) [5] was used.

Diarization "Who Spoke When" (SPKR): SPKR systems are required to annotate a meeting with regions of time indicating when each meeting participant is speaking and clustering the regions by speaker. It is a clustering task as opposed to an identification task since the system is not required to output a name for the speakers – only a generic id which is unique within the meeting excerpt being processed.

The Diarization Error Rate (DER) metric is used to assess SPKR system performance. DER is the ratio of incorrectly attributed speech time, (either falsely detected speech, missed detections of speech, or incorrectly clustered speech) to the total amount of speech time, expressed as a percentage. As with WER, a score of zero indicates perfect performance and higher scores indicate poorer performance.

In order to determine incorrectly clustered speech, the Hungarian solution to a bipartite graph[1] is used find a one-to-one mapping between the system-generated speaker segment clusters and the reference speaker segment clusters. Once the mapping is found, speech time within a system speaker cluster not matching speech time in the mapped reference speaker cluster is classified as the incorrectly clustered speech.

New for 2006, the primary measure of DER was calculated for all speech including overlapping speech. This harmonizes the scores with the STT task which now includes the evaluation of overlapping speech.

Inherent ambiguities in pinpointing speech boundaries in time and annotator variability result in a small degree of inconsistency in the time annotations in the reference transcript. To address this, a 0.25 second "no score" collar is created around each reference segment. This collar effectively minimizes the amount of DER error due to reference annotation inconsistencies.

Another challenge is in determining how large a pause in speech must be to cause a segment break. Although somewhat arbitrary, a cutoff value of 0.3 seconds was empirically determined to be a good approximation of the minimum duration for a pause in speech resulting in an utterance boundary. As such, segments that are closer than 0.3 seconds apart are merged in both the reference and system output transcripts.

The MDM audio input condition was the primary evaluation condition for the SPKR task for both meeting sub-domains. The *confmtg* data supported one contrastive condition, SDM, and the *lectmtg* data supported four contrastive conditions: SDM, MSLA, MM3A, and MBF. Participants could submit systems for the *confmtg* domain, the *lectmtg* domain, or both the sub-domains.

[1] http://www.nist.gov/dads/HTML/HungarianAlgorithm.html

Diarization "Speech Activity Detection" (SAD): SAD systems are required to annotate a meeting with regions of time indicating when at least one person is talking. The SAD task is therefore a simplified version of the SPKR task (because no speaker clustering is performed by the system). The task was introduced as an entry point for new participants in the RT evaluation series and to gauge the contribution of SAD errors to the SPKR and STT tasks.

The task was a dry run for the RT-05S evaluation but was considered a full evaluation task for RT-06S.

Since SAD is viewed as a simplification of the SPKR task, the SPKR DER scoring metric is also used to score the SAD task. The same no-score collar, 0.25 seconds, was applied during scoring and the same smoothing parameter, 0.3 seconds, was applied to the reference files. The reference files were derived from the SPKR reference files by simply merging the reference speaker clusters into a single cluster and then merging segments that either overlap or were within the 0.3 second smoothing parameter.

The MDM audio input condition was the primary evaluation condition for the SAD task for both meeting sub-domains. The *confmtg* data supported two contrastive conditions: SDM and IHM, and the *lectmtg* data supported five contrastive conditions: SDM, IHM, MSLA, MM3A, and MBF. Participants could submit system outputs for the *confmtg* domain, the *lectmtg* domain, or both sub-domains

The SAD task using IHM data is not directly comparable to SAD on distant microphone data, (i.e., MDM, SDM, MSLA, or MM3A data). An IHM channel includes both the wearer's speech and cross-talk from other meeting participants. For the purposes of this evaluation, this cross talk is not considered detectable speech even though it was human generated. Therefore, an IHM SAD system has the challenging task of detecting the primary speaker's speech and differentiating it from the cross-talk.

2.4 RT-06S Evaluation Corpora Details

As indicated previously, the RT-06S evaluation data consisted of two test sets: a conference room meeting test set and a lecture room meeting test set. The recordings were sent to participants as either down-sampled, 16-bit, 16Khz NIST Speech Header Resources (SPHERE) files or in the original 24-bit, 44.1 Khz WAV sample format as well as headerless raw files. The recordings of the meetings in the *confmtg* data set were distributed in their entirety while only the selected excerpts from the *lectmtg* data were distributed. All meeting recordings included video recordings. The video recordings were not distributed to the RT participants but they were distributed to the CLEAR evaluation participants.

Conference Room Meetings: The *confmtg* test set consisted of nominally 162 minutes of meeting excerpts from nine different meetings. NIST selected 18 minutes from each meeting to include in the test set. For two of the nine meetings, the excerpts were not contiguous. Five sites contributed data: the Augmented Multi-party Interaction (AMI) Project provided two meetings collected at Edinburgh University (EDI) and one meeting collected at TNO. Carnegie Mellon University (CMU), the National Institute of Standards and Technology (NIST), and Virginia Tech (VT) each contributed two meetings. The Linguistic Data Consortium (LDC) transcribed the test set according to the "Meeting Data Careful Transcription Specification - V1.2" guidelines [4]. Table 1 gives the salient details concerning the *confmtg* evaluation corpus.

Each meeting recording evaluation excerpt met minimum sensor requirements. Each meeting participant wore a head-mounted close talking microphone and there were at least three table-top microphones placed between the meeting participants. The dialects were predominately American English with the exception of the EDI meetings. In addition to these sensors, the EDI and TNO meetings included an eight-channel circular microphone array placed on the table between the meeting participants.

Table 1. Summary of Conference Room Meeting evaluation corpus

Meeting ID	Duration (minutes)	Number of Participants	Notes
CMU_20050912-0900	17.8	4	Transcription team mtg.
CMU_20050914-0900	18.0	4	Transcription team mtg.
EDI_20050216-1051	18.0	4	Remote control design
EDI_20050218-0900	18.2	4	Remote control design
NIST_20051024-0930	18.1	9	Project planning mtg.
NIST_20051102-1323	18.2	8	Data resource planning
TNO_20041103-1130	18.0	4	Remote control design
VT_20050623-1400	18.0	5	Problem solving scenario
VT_20051027-1400	17.7	4	Candidate selection
Total		46	
Unique speakers		43	

Lecture Room Meetings: The *lectmtg* test set consisted of 190 minutes of lecture meeting excerpts recorded at AIT, IBM, ITC-irst, and UKA. There were 38, 5-minute excerpts included in the evaluation from 26 recordings. Two types of lectures were recorded for the evaluation: "lectures" and "interactive lectures". Both lecture types were technical language technology talks given by invited lecturers. The lectures involved a single lecturer and a large group of audience members: only a few of which wore head microphones. In contrast, the interactive lectures were smaller groups and included not only the recording of the lecture but also people entering the room and coffee breaks. All participants in the interactive lectures wore head microphones. While the coffee breaks and person movements were useful for CLEAR evaluation, the data was unlike the data used for previous RT evaluations.

The excerpts were selected and transcribed by ELDA. After the evaluation, CMU revised the transcripts to include speech only recorded on the table-top microphones. During the revision, twelve of the excerpts were deemed to be of insufficient audio quality for the evaluation and removed from test set. This resulted in a 130-minute test set from 24 meetings.

The audio sensors used in the *lectmtg* data were configured differently than the *confmtg* data. Only the lecturer and two-to-four audience members, of potentially several, wore head-mounted, close-talking microphones. The rest of the audience was audible on the distant microphones. Three-to-four microphones were placed on the table in front of the lecturer and a fifth table-top microphone was placed in the corner of the room. Four-to-six source localization arrays were mounted on each of the four walls of the room. Finally, one or two Mark III arrays were placed directly in front of the lecturer.

3 Results of the RT-06S Evaluation

3.1 RT-06S Evaluation Participants

The following table lists the RT-06S participants and the evaluation tasks each site took part in. In total, there were ten sites submitting system outputs.

Table 2. Summary of evaluation participants and the tasks for which systems were submitted

Site ID	Site Name	Evaluation Task		
		STT	SPKR	SAD
AIT	Athens Information Technology		X	X
AMI	Augmented Multiparty Interaction Program	X	X	X
IBM	IBM	X		X
ICSI/SRI	International Computer Science Institute and SRI International	X	X	X
INRIA	Institut National de Recherche en Informatique et en Automatic			X
ITC-irst	Center for Scientific and Technological Research			X
LIA	Laboratoire Informatique d'Avignon		X	X
LIMSI	Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur	X	X	X
UKA	Karlsruhe University (UKA)	X		
UPC	Universitat Politècnica de Catalunya			X

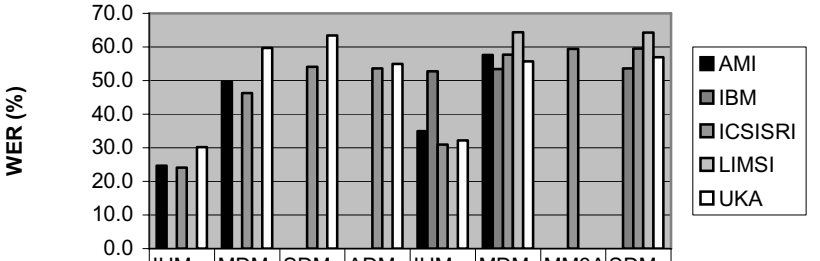
3.2 Speech-To-Text (STT) Results

Five sites participated in the STT task: AMI, IBM, ICSI/SRI, LIMSI, and UKA. This was the first year for IBM, LIMSI, and UKA. AMI, ICSI/SRI, and UKA submitted system outputs for the *confmtg* data while all sites submitted system outputs for the *lectmtg* data.

Figure 1 contains the results of all primary systems. The MDM WERs for *confmtg* data were 49.6, 46.3, and 59.7 for AMI, ICSI/SRI, and UKA respectively. The MDM WERs for the *lectmtg* data were 57.6, 53.4, 57.7, 64.4, and 55.7 for AMI, IBM, ICSI/SRI, LIMSI, and UKA. The *lectmtg* WERs for AMI and ICSI/SRI were 16% and 25% (relative) higher than *confmtg* data. However, UKA did 6% (relative) better on the *lectmtg* data. Last year, AMI and ICSI/SRI had higher error rates for the *lectmtg* data.

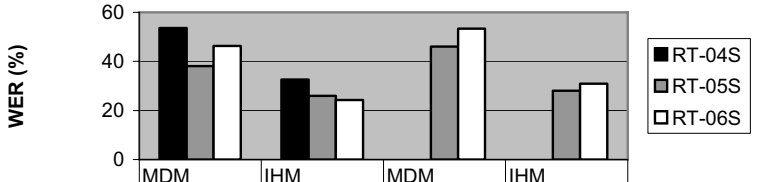
Unlike last year, the IHM error rates are higher for the *lectmtg* data: 41%, 28%, and 6% relative for AMI, ICSI/SRI and UKA respectively. One possible explanation for the increase is the dialect of the speakers. Many *lectmtg* speakers speak with strong, non-English accents, e.g., German- and Spanish-accented English.

A notable result from the *lectmtg* data was ICSI/SRI's 59.4% MM3A score. This result used the beamformed signal from built by UKA's Probabilistic Data Association Filters [10]. This was the first time in an RT evaluation that an automatic, blind source separation algorithm was applied to the output of Mark III arrays for use in aan STT system.



	IHM	MDM	SDM	ADM	IHM	MDM	MM3A	SDM
	Conference Room			Lecture Room				
AMI	24.7	49.6			35.0	57.6		
IBM					52.8	53.4		53.6
ICSISRI	24.1	46.3	54.1	53.6	31.0	57.7	59.4	59.5
LIMSI						64.4		64.3
UKA	30.2	59.7	63.4	55.0	32.2	55.7		57.0

Fig. 1. WERs for primary STT systems across test sets and audio input conditions. Up to 4 simultaneous speakers included in distant mic. conditions.



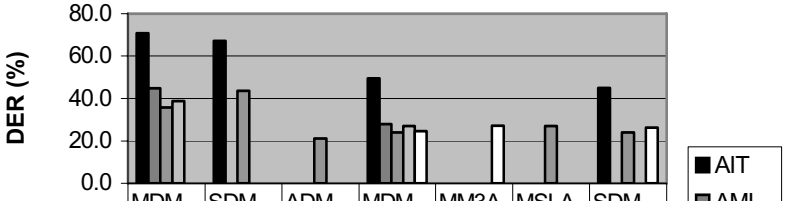
	MDM	IHM	MDM	IHM
	Conference Room		Lecture Room	
RT-04S	53.6	32.7		
RT-05S	38.1	25.9	46.1	28.1
RT-06S	46.3	24.2	53.4	31

Fig. 2. WERs for the best STT systems from RT-04S through RT-06S. MDM results are for segments with <= 4 active speakers while the IHM results include all speech.

Figure 2 plots the historical error rates for the MDM and IHM conditions in both domains. There was a slight reduction in IHM WERs for the *confmtg* data. However, both MDM and IHM error rates were higher for the RT-06 *lectmtg* data.

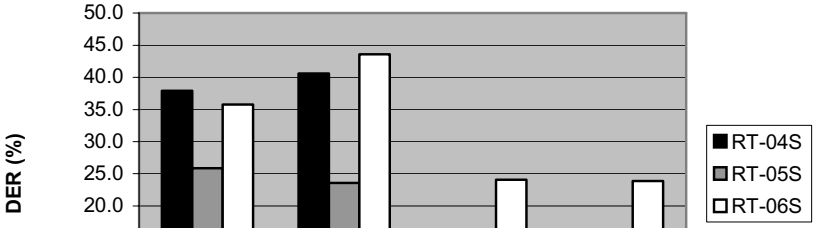
3.3 Diarization "Who Spoke When" (SPKR) Results

Five sites participated in the SPKR task: AIT, AMI, ICSI, LIA, and LIMSI. Of the five, only ICSI participated in the RT-05S SPKR evaluation. Figure 3 contains the results of all primary systems. The MDM DERs for *confmtg* data were 70.7, 44.8, 35.8, and 38.8 for AIT, AMI, ICSI, and LIA respectively. The MDM DERs for the *lectmtg* data were 49.5, 27.8, 24.0, 27.0, and 24.6 for AIT, AMI, ICSI, LIA, and LIMSI.



	MDM	SDM	ADM	MDM	MM3A	MSLA	SDM
	Conference Meeting		Lecture Meeting				
AIT	70.7	67.2		49.5			45.0
AMI	44.8			27.8			
ICSI	35.8	43.6	21.1	24.0		27.0	24.0
LIA	38.8			27.0			
LIMSI				24.6	27.1		26.2

Fig. 3. DERs for the all speech for the primary SPKR systems across test sets and audio input conditions



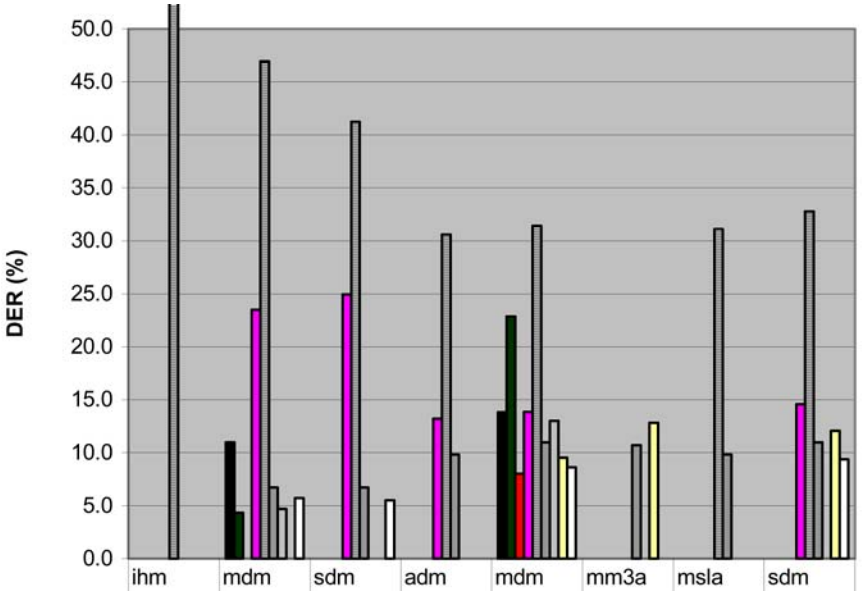
	MDM	SDM	MDM	SDM
	Conference Room		Lecture Room	
RT-04S	37.9	40.6		
RT-05S	25.9	23.6	12.5	12.3
RT-06S	35.8	43.6	24.1	23.9

Fig. 4. DERs for the best MDM and SDM SPKR systems from RT-04S through RT-06S

The scores were appreciably higher then last year. Figure 4 contains the historical lowest error rates for each year. There are a couple of factors that may have attributed to the increase. First, these are all new systems. Second, the reference file generation continued to be problematic. Using human segmentation annotations are problematic in that consistency is hard to achieve. Future evaluations will use reference files derived from word-level forced alignments of reference transcription.

3.4 Diarization "Speech Activity Detection" (SAD) Results

Nine sites participated in this first formal evaluation SAD task during the RT evaluation: AIT, AMI, IBM, ICSI, INRIA, ITC-irst, LIA LIMSI, and UPC. Figure 5 shows the lowest MDM error rate for the *confmtg* data was achieved by AMI with a DER of



AIT		11.0			13.8			
AMI		4.3			22.8			
IBM					8.0			
ICSI		23.5	25.0	13.2	13.8			14.6
INRIA	78.5	47.0	41.2	30.6	31.4		31.1	32.8
ITC		6.7	6.7	9.8	11.0	10.7	9.8	11.0
LIA		4.7			13.0			
LIMSI					9.5	12.9		12.1
UPC		5.7	5.5		8.6			9.4

Fig. 5. DERs for primary SAD systems across test sets and audio input conditions

4.3%. For the *lectmtg* data, IBM had the lowest MDM DER of 8.02%. As with the other tasks, the *lectmtg* data had higher error rates than the *confmtg* data.

The SAD task will be continued in future evaluations since both the Dry Run in 2005 and the evaluation in 2006 were successful.

4 Conclusions and Future Evaluations

The collaboration between RT and the CLEAR evaluation as well as the AMI, CHIL and VACE programs has boosted the RT community on many levels. For the first time, the RT evaluation corpora has been annotated and used for the evaluation of both language and video processing/extraction tasks. The collaboration has also led to expanded task participation: almost twice the number of systems were built by the participants even though the number of participating sites remained constant. We look forward to continued progress and evaluations in the meeting domain.

The RT-07 evaluation is being planned for the Spring of 2007. As with 2006, the same evaluation corpora will be used for both RT and CLEAR. In addition, the RT and CLEAR evaluation workshops will be co-located.

Applying blind source separation techniques RT is an exciting new direction for RT systems. We anticipate further sensor fusion will be possible as the CLEAR and RT communities are merged.

Acknowledgements

NIST would like to thank AIT, EDI, CMU, IBM, ITC, VT, UKA, TNO, and UPC for donating meeting recordings to the evaluation. Special thanks go to CMU, ELDA, and LDC for preparing reference transcriptions and annotations. The authors thank and appreciate the edits provided by Vince Stanford.

Disclaimer

These tests are designed for local implementation by each participant. The reported results are not to be construed, or represented, as endorsements of any participant's system, or as official findings on the part of NIST or the U. S. Government. Certain commercial products may be identified in order to adequately specify or describe the subject matter of this work. In no case does such identification imply recommendation or endorsement by NIST, nor does it imply that the products identified are necessarily the best available for the purpose.

References

1. Fiscus et. al., "Results of the Fall 2004 STT and MDE Evaluation", RT-04F Evaluation Workshop Proceedings, November 7-10, 2004.
2. Garofolo et. al., "The Rich Transcription 2004 Spring Meeting Recognition Evaluation", ICASSP 2004 Meeting Recognition Workshop, May 17, 2004

3. Spring 2006 (RT-06S) Rich Transcription Meeting Recognition Evaluation Plan, http://www.nist.gov/speech/tests/rt/rt2006/spring/
4. LDC Meeting Recording Transcription, http://www.ldc.upenn.edu/Projects/Transcription/NISTMeet
5. SCTK toolkit, http://www.nist.gov/speech/tools/index.htm
6. Michel et. al., "The NIST Meeting Room Phase II Corpus", 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI-06), 1-3, May 2006.
7. Fiscus et. al., "The Rich Transcription 2005 Spring Meeting Recognition Evaluation", The joint proceedings Rich Transcription Workshop and the 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI), 11-13 July 2005
8. http://www.clear-evaluation.org/
9. Fiscus et. al., "Multiple Dimension Levenshtein Distance Calculations for Evaluating Automatic Speech Recognition Systems During Simultaneous Speech", LREC 2006: Sixth International Conference on Language Resources and Evaluation
10. Gehrig and McDonough, "Tracking Multiple Simultaneous Speakers with Probabilistic Data Association Filters", 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI-06)
11. Stanford, V.: The NIST Mark-III microphone array - infrastructure, reference data, and metrics. In: Proceedings International Workshop on Microphone Array Systems - Theory and Practice, Pommersfelden, Germany (2003)
12. http://isl.ira.uka.de/clear06/downloads/ClearEval_Protocol_v5.pdf

The IBM RT06s Evaluation System for Speech Activity Detection in CHIL Seminars

Etienne Marcheret, Gerasimos Potamianos,
Karthik Visweswariah, and Jing Huang

IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
{etiennem,gpotam,kv1,jhgh}@us.ibm.com

Abstract. In this paper, we describe the IBM system submitted to the NIST Rich Transcription Spring 2006 (RT06s) evaluation campaign for automatic speech activity detection (SAD). This SAD system has been developed and evaluated on CHIL lecture meeting data using far-field microphone sensors, namely a single distant microphone (SDM) configuration and a multiple distant microphone (MDM) condition. The IBM SAD system employs a three-class statistical classifier, trained on features that augment traditional signal energy ones with features that are based on acoustic phonetic likelihoods. The latter are obtained using a large speaker-independent acoustic model trained on meeting data. In the detection stage, after feature extraction and classification, the resulting sequence of classified states is further collapsed into segments belonging to only two classes, speech or silence, following two levels of smoothing. In the MDM condition, the process is repeated for every available microphone channel, and the outputs are combined based on a simple majority voting rule, biased towards speech. The system performed well at the RT06s evaluation campaign, resulting to 8.62% and 5.01% "speaker diarization error" in the SDM and MDM conditions respectively.

1 Introduction

Speech activity detection (SAD) has long been an important issue as a front end step to the *automatic speech recognition* (ASR) process. Its significance ranges (although not limited to) from bandwidth usage in the client/server ASR paradigm, to stable prompt control during barge-in operation. SAD has a positive impact on ASR in terms of both CPU usage and accuracy, since the decoder is not required to operate on non-speech segments, reducing processing effort and word insertion error rate. Furthermore, robust performance of SAD is crucial in developing technologies for the smart room domain, for example lecture seminars and meetings, as in the CHIL ("Computers in the Human Interaction Loop") project [1]. There, in addition to ASR, SAD is useful as a pre-processing step for speaker localization.

Not surprisingly, speech activity detection has attracted significant interest in the ASR literature. Most techniques are based on features extracted from the acoustic signal, ranging from energy [2] to frequency-based representations of speech [3, 4, 5]. The selected features are subsequently used in speech/ silence classification, ranging from adaptive thresholding to linear discriminants,

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 323–335, 2006.

regression trees, distance measures, and Gaussian mixture model (GMM) based classifiers. In general, energy-based SAD is computationally efficient and simple to implement, but it lacks robustness to noise. Performance can be improved by using adaptive thresholds or appropriate filtering of the energy estimates [2, 6], however addressing non-stationary noise effectively remains difficult. Most often, frequency-based speech features, such as mel-frequency cepstral coefficients (MFCCs), are required to achieve improved robustness to noise.

In previous work [7, 8], we have proposed to employ such MFCC features indirectly, through the acoustic model that is assumed to generate them. The resulting acoustic phonetic features were extracted based on the phonetic class conditional observation vector likelihoods by the acoustic model, and were used to augment traditional energy-like based features. The two types of features were subsequently fused with a simple concatenation and a projection-based dimensionality reduction, and fed to a Gaussian mixture classifier for speech/silence detection. Among other domains, the proposed algorithm was tested on single-microphone, far-field acoustic data, collected as part of the CHIL project [8], during the first CHIL-internal evaluation campaign ("CHIL eval. run #1"), achieving excellent results.

Since then, a number of slight modifications have been applied to the IBM SAD system in an effort to improve its performance for the "Rich Transcription Spring 2006" (RT06s) evaluation campaign. In particular, the diagonal covariance GMM classifier was replaced by a full covariance model operating on a reduced set of features, by eliminating highly correlated features. In addition, the acoustic model has been replaced by a PLP-feature based model developed for the RT06s speech-to-text evaluation [9]. Finally, to allow operation on multiple microphone inputs, a simple majority voting scheme has been implemented to combine single-microphone SAD system outputs. This corresponds to channel integration (fusion) in the "decision" level, instead of the "signal" level followed by other works in the literature [10, 11].

Our SAD system improvements are discussed in detail in Sections 2, 3, and 4. In particular, Section 2 is devoted to feature extraction, Section 3 to SAD system training, and Section 4 focuses on SAD testing. A number of development experiments and SAD evaluation results are presented in Section 5. Finally, Section 6 concludes the paper with a short summary.

2 Feature Extraction for SAD

As discussed in the Introduction, the IBM SAD system operates on two types of features: Energy based ones, generated directly from the waveform, and acoustic phonetic features, defined from observations generated by the ASR acoustic model. The two feature sets are combined, and are subsequently fed to a Gaussian mixture model (GMM) classifier, as discussed in Section 3.

2.1 Energy Based Features

The energy based feature space is defined by a five-dimensional vector, the components of which are based on the bandpass filtered acoustic waveform within

the [200, 900] Hz range. Letting $y[i]$ denote the bandpass-filtered waveform at sample time i, the estimated short time energy $e(t)$ for a window of length N is given by

$$e(t) = 10 \log \left(\frac{1}{N} \sum_{i=1}^{N} y[i]^2 \right) , \tag{1}$$

measured in dB. In (1), t is discrete and determined by the observation frame rate, set in this work to be every 10 ms. This results to $N = 160$ samples in (1) for 16 kHz audio. Given $e(t)$, we generate filtered observations of it, based on

$$rms(t) = 10^{\, scale \times \, e(t)} . \tag{2}$$

In (2), $rms(t)$ is defined as a linear energy, scaled for the expected number of bits of resolution. The scaling constant is given by $scale = contrast/scaleMax$, where the *contrast* provides a level of sensitivity, generally set within [3.5, 4.5], and $scaleMax$ is the maximum possible value of $e(t)$, for example 90.3 dB for a 16-bit signed linear PCM signal.

Based on the instantaneous $rms(t)$ value, we can obtain "low", "mid", and "high" energy tracks, defined as

$$lt(t) \;=\; (1 - \alpha_{l,t}) \times lt(t-1) \;+\; \alpha_{l,t} \times rms(t) \tag{3a}$$

$$mt(t) \;=\; (1 - \alpha_{m}) \times mt(t-1) \;+\; \alpha_{m} \times rms(t) \tag{3b}$$

$$ht(t) \;=\; (1 - \alpha_{h,t}) \times ht(t-1) \;+\; \alpha_{h,t} \times rms(t) \tag{3c}$$

respectively. For the mid-track $mt(t)$, time constant α_m is fixed, set in this work to 0.1. Therefore $mt(t)$ is the lowpass filtered $rms(t)$. The remaining low and high track time constants $\alpha_{l,t}$ and $\alpha_{h,t}$ are functions of the instantaneous $rms(t)$, designed so that rapid changes in energy will cause abrupt tracking by $lt(t)$ and $ht(t)$, respectively. They are given by

$$\alpha_{l,t} = \left(\frac{lt(t-1)}{rms(t)} \right)^2 \quad \text{and} \quad \alpha_{h,t} = \left(\frac{rms(t)}{ht(t-1)} \right)^2 ,$$

thus resulting in increasing $\alpha_{l,t}$ for decreasing $rms(t)$, and increasing $\alpha_{h,t}$ for increasing $rms(t)$.

Next, from (3), we form three equivalent low, mid, and high energy representations, given by

$$let(t) = \frac{\log\left(lt(t)\right)}{scale} , \quad met(t) = \frac{\log\left(mt(t)\right)}{scale} , \quad \text{and} \quad het(t) = \frac{\log\left(ht(t)\right)}{scale} . \tag{4}$$

From (4), we can also obtain the mid-to-low energy track relationship as

$$m2l(t) = met(t) - let(t) . \tag{5}$$

By combining (1), (4), and (5) we obtain a five-dimensional energy feature vector at frame t, as

$$v_e(t) = [\, e(t) \;\; let(t) \;\; met(t) \;\; het(t) \;\; m2l(t) \,] . \tag{6}$$

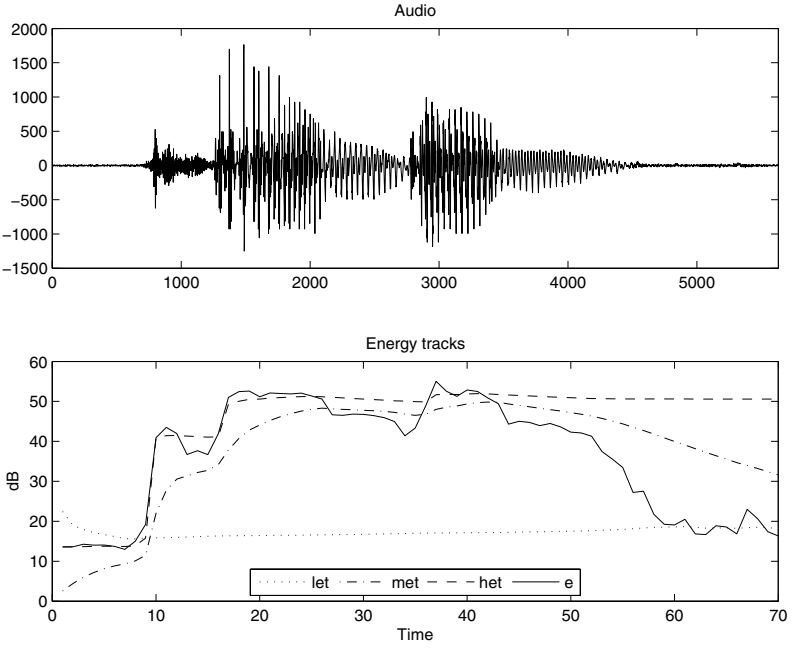


Fig. 1. Audio waveform and corresponding energy tracks

A purely energy based speech activity detector based on these observations where dynamic speech/silence thresholds are employed can be found in [6].

Fig. 1 shows the bandpass filtered energy (1) and the corresponding energy tracks *let*, *met*, and *het* defined in (4). We observe that the low and high energy tracks are intended to lock onto the "floor" and speech signal levels respectively, while the mid track is a lowpass filtered energy track.

2.2 Acoustic Phonetic Features

The acoustic phonetic feature space employed for speech activity detection is derived from the acoustic model used for ASR. The acoustic model is generated from partitioning the acoustic space by context-dependent phonemes with the context defined in this work as plus and minus five phonemes, cross-word to the left only. The context-dependent phoneme observation generation process is modeled as a GMM within the hidden Markov model (HMM) framework, and in typical large-vocabulary ASR systems, this can easily lead to more than 1k states and 40k Gaussian mixture components. Calculating all HMM state likelihoods from all Gaussians at each frame would preclude real-time operation. Therefore, we define a hierarchical structure for the Gaussians, where it is assumed that only a small subset of them is significant to likelihood computation at any given time [12]. The hierarchical structure takes advantage of the sparseness by surveying the Gaussian pool in multiple resolutions given some acoustic feature vector $\mathbf{x}$. As part of the training process, the complete set of available Gaussian densities

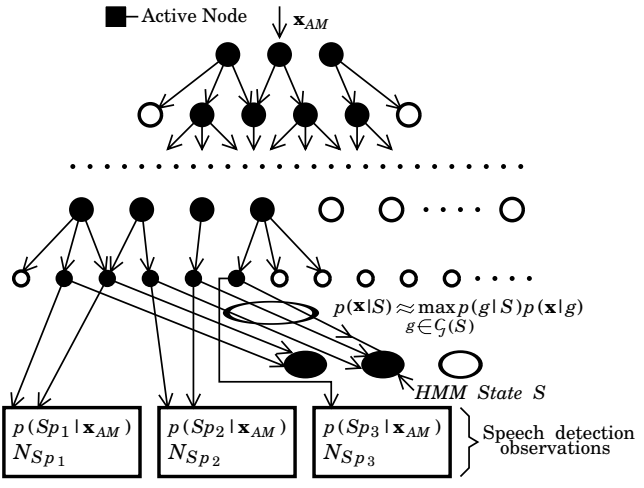


Fig. 2. Hierarchical acoustic model and the corresponding acoustic phonetic speech detection observations

is clustered into a search tree, in which the leaves correspond to the individual Gaussians, and a parent node is the centroid of its children for a defined distance metric. At the bottom of this tree resides a many-to-one mapping, collapsing the individual Gaussians to the appropriate HMM state. Therefore, the HMM state s conditional likelihood of a given observation vector $\mathbf{x}$ at time t is computed as

$$p(\mathbf{x}|s) = \sum_{g \in \mathcal{G}(s)} p(g|s)\, p(\mathbf{x}|g) \,,$$

where $\mathcal{G}(s)$ is the set of Gaussians that make up the GMM for state s. Traversing the tree yields a subset of active Gaussians, denoted by $\mathcal{Y}$. Based on $\mathcal{Y}$ and the many-to-one mapping, the conditional likelihood of a state is approximated as

$$p(\mathbf{x}|s) = \max_{g \in \mathcal{Y} \cap \mathcal{G}(s)} p(g|s)\, p(\mathbf{x}|g) \,.$$

If no Gaussian from a state is present in $\mathcal{Y}$, a default floor likelihood is assigned to that state.

To define the acoustic phonetic space used for speech activity detection, we apply an additional many-to-one mapping to the pruned result of the hierarchical tree. The mapping is based on grouping phonemes into three broadly defined classes: (i) the pure silence phoneme, trained from non-speech; (ii) the disfluent phonemes, which are noise-like phonemes, namely the unvoiced fricatives and plosives, i.e., the ARPAbet subset {/b/, /d/, /g/, /k/, /p/, /t/, /f/, /s/, /sh/}; and (iii) all the remaining phonemes, such as the vowels and voiced fricatives. The three classes will be denoted by Sp_1, Sp_2, and Sp_3, respectively. Then, from the acoustic feature $\mathbf{x}$, used to traverse the acoustic model hierarchy, we can

form the speech detection class posteriors for the three speech detection classes as

$$Pr(Sp_i|\mathbf{x}) \;=\; \frac{1}{acc\_mass} \sum_{g \,\in\, \mathcal{Y} \cap \mathcal{G}(Sp_i)} p(\mathbf{x}|g)\; p(g|Sp_i)\;, \tag{7}$$

where

$$acc\_mass = \sum_{i=1}^{3} \left\{ \sum_{g \,\in\, \mathcal{Y} \cap \mathcal{G}(Sp_i)} p(\mathbf{x}|g)\; p(g|Sp_i) \right\},$$

and $\mathcal{G}(Sp_i)$ is the set of Gaussians defined by the mapping from phoneme to speech detection class Sp_i .

The process is illustrated in Fig. 2. Notice that the pruning at each level is accomplished using a threshold relative to the maximum scoring likelihood for that level [12]. As a result, the sharper the drop-off in Gaussian likelihoods, the more aggressive the pruning becomes. Therefore, both SNR and the phoneme being pronounced impacts the pruning. Features extracted from vowels and other voiced phonemes will result in more aggressive pruning than unvoiced fricatives, plosives and silence phonemes. This pruning will remain relative to SNR, with increasing SNR resulting in an overall more aggressive pruning.

The above observation results in additional speech detection features, based on class-normalized Gaussian counts. Denoting by N_{Sp_i} the number of Gaussians after hierarchical pruning that map to speech detection class Sp_i (see also Fig. 2), we consider the normalized counts

$$\overline{N}_{Sp_i} = N_{Sp_i} \,/\, \sum_{j=1}^{3} N_{Sp_j}\;, \quad \text{for} \quad i = 1, 2, 3\,, \tag{8}$$

as additional features. Combining (7) and (8) we obtain the six-dimensional acoustic phonetic feature space at frame t given by $v_a(t)$, as defined in (9):

$$\begin{aligned} v_{ai}(t) &= \left[\, log\left(Pr(Sp_i|\mathbf{x})\right) \;\; log\left(\overline{N}_{Sp_i}\right) \right] \\ v_a(t) &= \left[\, v_{a1}(t) \;\; v_{a2}(t) \;\; v_{a3}(t) \,\right]\;. \end{aligned} \tag{9}$$

3 SAD System Training

The SAD system training consists of two steps: the first step concerns training the acoustic model, whereas the second focuses on training the speech/silence classifier. Details are provided in the following subsections.

3.1 Acoustic Model Training

In order to generate the acoustic phonetic features described in Section 2.2, an acoustic model is required. Such a model is trained based on far-field lecture and meeting data, as described in an accompanying paper [9]. To summarize, this is a speaker-independent model based on 40-dimensional features generated

from an LDA projection, applied to a concatenation of nine consecutive 13-dimensional PLP acoustic observation vectors. Such observations are computed at 100 Hz from a Hamming windowed 25 ms speech segment, and are mean normalized on a per-speaker basis. The resulting acoustic model is composed of three-state, left-to-right HMM phonetic models, with the final model having a total of 6000 context-dependent states and approximately 200k Gaussians. The model is trained on 473.5 hrs of far-field data [9].

3.2 Speech/Silence Classifier Training

The fundamental classifier employed for speech/silence detection is a Gaussian mixture model (GMM). In this work, we investigate two modeling approaches, namely a diagonal GMM and a full covariance GMM. Details are provided next.

Feature Combination for the Diagonal GMM: From (6) and (9) we derive fused (concatenated) 11-dimensional features

$$v_f(t) = [\, v_e(t) \;\; v_a(t) \,] \; . \tag{10}$$

In order to de-correlate such features and allow classification by a diagonal-covariance GMM, we apply *principal component analysis* (PCA) to (10). In particular, we choose as subspace the basis set formed by the eigenvectors corresponding to the top eight eigenvalues. This results in the projected eight-dimensional feature vector

$$v_p(t) = \mathbf{A}\, v_f(t) \; , \tag{11}$$

where $\mathbf{A}$ denotes the PCA matrix.

Feature Combination for the Full Covariance GMM: In order to accomplish full-covariance GMM training, features with high correlation need to be removed from the observation vector. In particular, in (6), we drop the mid-to-low energy track $m2l(t)$, yielding a four-dimensional energy based feature vector

$$v_{e,FC}(t) = [\, e(t) \;\; let(t) \;\; met(t) \;\; het(t) \,] \; .$$

Concerning the acoustic-phonetic features, it may not be immediately obvious from (9) that each observation pair in (7) and (8) are highly correlated. Keeping all features results in sharp models that don't generalize robustly. It was experimentally determined that class-normalized Gaussian counts (see (8)) outperform posteriors (7) as features for SAD. Therefore, the following seven-dimensional feature vector is used in conjunction with the full covariance GMM:

$$v_{FC}(t) = [\, v_{e,FC}(t) \;\; log\left(\overline{N}_{Sp_1}\right) \;\; log\left(\overline{N}_{Sp_2}\right) \;\; log\left(\overline{N}_{Sp_3}\right) \,] \; . \tag{12}$$

GMM Training: We subsequently train a three-class GMM classifier (for each speech detection class described in Section 2.2) on vectors (11) in the diagonal GMM case, or (12) in the full covariance case. GMM training is accomplished in two steps: First, we pool all training vectors – independently of the associated

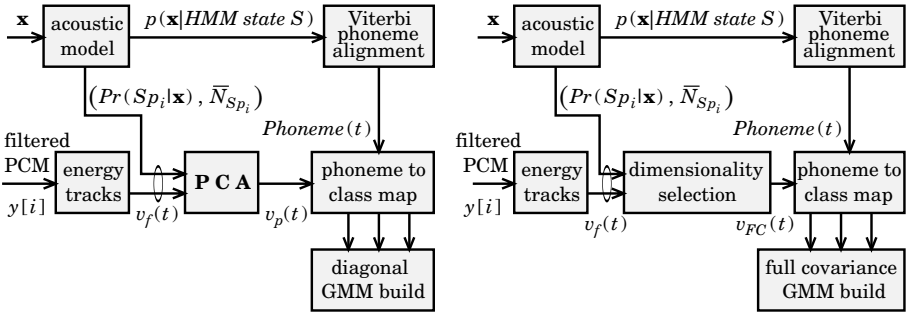


Fig. 3. Training the three-class speech detection classifier using diagonal (left) or full-covariance (right) GMMs

class labels – and run the LBG splitting algorithm [13], where the *expectation maximization* (EM) algorithm is iteratively run to convergence between splits. Splitting is terminated when the desired number of mixtures is reached: That number is eight in the case of the diagonal GMM, and two or four mixtures in the case of the full covariance models, as discussed in our experiments (Section 5). This initial step generates a class-independent GMM. Subsequently, the training vector class labels are obtained by Viterbi alignment, using the acoustic model and the phoneme-to-speech detection class mapping (see also Fig. 3). A single EM step is then run employing the individually pooled class-specific training vectors, with the class-independent GMM used as the starting model. The effectiveness of this process has been determined empirically and borrows from techniques found in speaker verification literature [14].

In particular, the GMMs were trained using the following resources: (i) All CHIL 2006 development data; (ii) All CHIL data within the RT05 development set; and (iii) Downsampled ICSI, NIST, RT04, non-CHIL RT05, and AMI data (see also [9]). Far-field microphone audio data were pooled together, resulting in approximately 19 hours of CHIL data and 4.7 hours from the additional sets.

4 SAD System Testing

Following the acoustic model and GMM training steps, we now proceed to describe how the SAD system is applied on single- and multi-channel audio signal inputs. Clearly, the first step is to extract the energy and acoustic features as described in Sections 2.1 and 2.2. Subsequently, the GMM is applied at each frame $t = n$ to features (11) or (12), for the diagonal or full covariance GMMs respectively. This results to the log-probability scores $L(Sp_i, n) = \log Pr(Sp_i|v_f(n))$, for each of the three classes discussed in Section 2.2. To obtain the final speech/silence intervals, these scores are further processed in the temporal domain by two stages of smoothing, the first of which also collapses the three possible classes into the two classes of interest: speech and silence. Finally, a third stage of processing is applied in the multiple distant microphone (MDM) evaluation condition. More details on these processing steps are given in the following.

4.1 Score Smoothing, Class Merging, and Score Integration

First, at each frame and for each of the three classes, we locally smooth scores $L(Sp_i, n)$ over a small fixed window of $w = 10$ consecutive frames (i.e., a window of 100 ms duration). The resulting smoothed scores are

$$\overline{L}(Sp_i, n) = \frac{1}{w} \sum_{k=0}^{w-1} L(Sp_i, n-k) \; .$$

We subsequently re-assign the "silence" class score at each frame, by merging the smoothed scores of the pure silence and disfluent classes by a simple max rule: $Score(silence, n) = \max_{i=1,2} \overline{L}(Sp_i, n)$. On the other hand, the "speech" score is initially set to the purely voiced class: $Score(speech, n) = \overline{L}(Sp_3, n)$. Following this step, the two scores are temporally accumulated as described next: When the global state is in "silence", the condition $Score(speech, n) > Score(silence, n)$ results in accumulation of the scores to determine if the global state is to be switched to speech. Denoting by frame n^* the initial frame in the silence state such that $Score(speech, n^*) > Score(silence, n^*)$, we begin integrating the difference between the "speech" and "silence" scores, namely

$$\Delta_{Sp}(n^* + \delta) = \frac{1}{\delta + 1} \sum_{k=0}^{\delta} \left(\overline{L}(Sp_3, n^* + k) - \max_{i=1,2} \overline{L}(Sp_i, n^* + k) \right) \; . \qquad (13)$$

The global state is then changed to "speech", once condition $\Delta_{Sp}(n^* + \delta) > 0$ is satisfied for any value $\delta \in [\, N_{min}\,, N_{max}\,]$. A similar procedure is used to switch from the "speech" to "silence" state, with the "speech" and "silence" scores swapped in (13). In our work, N_{min} and N_{max} are set to 50 and 100 ms respectively.

The above algorithm produces a segmentation where disfluent speech is lumped into the "silence" class. The last step is then to refine this segmentation in order to more accurately determine the true speech and silence boundaries. From the smoothed scores $\overline{L}(Sp_2, n)$, we know which regions were assigned to "silence" based on the disfluent class. Therefore, whenever a segment is classified as class Sp_2, it is mapped to speech (Sp_3), only if it lies between segments Sp_1 (silence) to its left and Sp_3 (speech) to its right (the condition of changing the global state from silence to speech), or vice-versa (speech to silence transition region); otherwise it is mapped to silence (Sp_1). This is intended to handle consonant-vowel-consonant transitions within a word, while maintaining robustness to non-stationary noise.

4.2 Lead, Lag, and Silence-Collapsed Smoothing

The above procedure provides a first signal segmentation into speech and silence intervals. However, this tends to be "over-segmented", and with very tight boundaries of the speech intervals. This necessitates a second level of smoothing that is driven by two temporal parameters, optimized based on development experiments. The first parameter, P_1, is designed to expand the speech intervals by P_1 ms on either side. Following such padding, a second parameter, P_2, is used to "collapse" silence segments that are of duration less than P_2 ms.

Table 1. "Speaker diarization error", %, for single-channel (SDM) SAD on three development sets for various SAD GMMs and smoothing parameters P_1 and P_2

SAD system parameters				development sets		
GMM covariance	# mixtures	P_1 (ms)	P_2 (ms)	Dev_1	Dev_2	Dev_3
Diagonal	8	300	150	9.77	1.10	9.25
Diagonal	8	300	250	9.62	1.17	9.12
Full	2	300	150	9.94	2.27	9.49
Full	2	300	250	9.69	1.92	9.25
Full	4	300	150	9.02	1.46	9.02
Full	4	300	250	8.84	1.23	8.94

4.3 SAD System Combination in the MDM Condition

It is expected that, when multiple microphone signals are available, speech activity detection may become more robust. This of course requires an appropriate fusion approach to combine the available multi-channel information. In this work, we choose a "decision fusion" methodology that utilizes class-only information, namely the single-channel SAD system outputs. In particular, for each time frame, we consider a simple *majority rule*, applied on the set of all available microphone channel SAD outputs at the particular time frame. The rule is implemented to be biased towards speech in case of a tie, which of course can only occur if the number of available microphone channels is even. Notice that this approach assumes synchronicity among all channels, which in general holds for the CHIL data due to the data capture mechanism and the relatively small smart room size.

Based on the majority rule, we consider two variants of SAD system combination in conjunction with the second level of SAD output smoothing that was discussed earlier (Section 4.2). In the first approach, referred to as "*Rover A*" method, smoothing is first applied independently per channel (as in the single-channel SAD system), followed by the majority rule decision. The second variant employs the first level of smoothing as in single-channel SAD (Section 4.1), but interjects the majority rule multi-channel fusion before applying the second smoothing stage (Section 4.2) to the combined output. The latter will be referred to as the "*Rover B*" method. "Rover A" is the technique used in the IBM MDM SAD system submitted to the RT06s evaluation campaign.

5 Experimental Results

We now proceed to report SAD system experimental results. We first summarize system variant comparisons on development data, followed by evaluation results achieved on the RT06s campaign.

In the previous sections, we discussed a number of possibilities for SAD system training and testing. For example, diagonal or full-covariance GMMs, parameters P_1 , P_2 for SAD output smoothing, and two "Rover" variants for multi-channel

Table 2. SAD speaker diarization error, %, on development data, when using two channel combination techniques on the multiple distant microphone (MDM) condition. Results on a single-channel (SDM) are also depicted. In all cases, a four-mixture, full-covariance GMM with $P_1 = 300$ ms and $P_2 = 250$ ms is used.

Condition / Method	Dev_1	Dev_2	Dev_3
SDM	8.84	1.23	8.94
MDM – "Rover A"	8.61	0.56	8.57
MDM – "Rover B"	8.72	0.53	8.52

Table 3. RT06s evaluation results for the IBM SDM and MDM SAD systems. Speaker diarization error, % (as well as detailed FA/FR, %), using the initial ELDA and final CMU reference transcripts are depicted.

Condition	Total err. (FA/FR) (ELDA ref.)	Total err. (FA/FR) (CMU ref.)
SDM	12.15 (5.7/6.5)	8.62 (2.3/5.3)
MDM	8.02 (2.8/5.2)	5.01 (0.2/4.2)

combination. To choose the optimal approach, we conduct a number of experiments on development data. We utilize three sets for this purpose:

- Set Dev_1: This set consists of seven seminars with a refined manual segmentation, kindly provided to us by CHIL partner UPC. The data include three seminars recorded at CHIL partner UKA, and one each at the IBM, ITC, AIT, and UPC sites. The segmentation is based on the original transcripts provided by CHIL partner ELDA, but only the UKA subset has been fully re-transcribed.
- Set Dev_2: This is a subset of the previous set, consisting of the three fully re-transcribed UKA seminars. The speech/silence reference segmentation is therefore quite accurate.
- Set Dev_3: This contains all CHIL dev. 2006 data, as segmented based on the ELDA transcripts.

It is important to note that the ELDA transcripts are well suited for ASR, but unfortunately are unreliable for SAD, as the speech/silence boundaries are not always accurately labeled. Thus, among the three development sets, only Dev_2 results can be fully trusted. However, the need for taking into account data from the other CHIL recording sites, when making system design decisions, forces us to also place emphasis on Dev_1 and Dev_3, so as to avoid overfitting to UKA data characteristics.

A number of development set results are depicted in Tables 1 and 2. In Table 1, we depict development set results for three GMM models, one using eight diagonal-covariance mixtures employing eight-dimensional features, the other with two or four full-covariance mixtures using seven-dimensional features (see also Section 3.2). The three models require 136, 74, and 146 parameters to be es-

timated, respectively. Based on this table, the four-mixture full-covariance model is chosen for the RT06s evaluation, since it achieves the best performance on two of the three sets, Dev_1 and Dev_3. In addition, parameter values $P_1 = 300$ ms and $P_2 = 250$ ms are chosen, based on a number of experiments that are not reported here due to lack of space.

Next, we compare the two channel combination techniques discussed in Section 4.3 for MDM SAD. Results are depicted in Table 2 for the chosen SAD system parameters. Single-channel SAD results are also depicted for comparison purposes. Clearly, improvements are significant, especially for set Dev_2 that is accompanied by the most accurate reference segmentation. However, no significantly consistent difference can be observed between the two combination techniques. "Rover A" is finally the technique chosen for the RT06s evaluation.

Finally, the RT06s evaluation results achieved by the IBM SDM and MDM systems are depicted in Table 3 of the CHIL eval. 2006 data set. Results using two reference segmentations are depicted: The ones derived from the initial ELDA transcripts, as reported in the RT06s workshop in May 2006, and the finalized segmentation after re-transcription by a team from the Carnegie Mellon University (CMU) in early June 2006. The latter are the official evaluation results. Note the significant improvement in the MDM condition, compared to the SDM results.

6 Summary

In this paper, we presented a novel approach to speech activity detection that augments energy based features with acoustic phonetic ones. In contrast to traditional systems that often use the frequency based speech representation to directly provide features to a speech/silence classifier, the proposed technique utilizes an acoustic model to provide likelihood based features to the detector using a phoneme grouping into three clusters. The algorithm performed well in the RT06s evaluation campaign, where it was applied to speech activity detection based on both single- and multi-channel far-field input.

Acknowledgements

The authors would like to acknowledge support of this work by the European Commission under integrated project CHIL, "Computers in the Human Interaction Loop", contract number 506909.

References

[1] Macho, D., Padrell, J., Abad, A., et al., "Automatic speech activity detection, source localization, and speech recognition on the CHIL seminar corpus," *Proc. ICME*, 2005.

[2] Li, Q., Zheng, J., Zhou, Q., and Lee, C.-H., "A robust, real-time endpoint detector with energy normalization for ASR in adverse environments," *Proc. ICASSP*, pp. 233–236, 2001.

[3] Martin, A., Charlet, D., and Mauuary, L., "Robust speech/non-speech detection using LDA applied to MFCC," *Proc. ICASSP*, pp. 237–240, 2001.
[4] Bou-Ghazale, S. and Assaleh, K., "A robust endpoint detection of speech for noisy environments with application to automatic speech recognition," *Proc. ICASSP*, pp. 3808–3811, 2002.
[5] Padrell, J., Macho, D., and Nadeu, C., "Robust speech activity detection using LDA applied to FF parameters," *Proc. ICASSP*, vol. 1, pp. 557–560, 2005.
[6] Monkowski, M., *Automatic Gain Control in a Speech Recognition System*, U.S. Patent US6314396.
[7] Marcheret, E., Visweswariah, K., and Potamianos, G., "Speech activity detection fusing acoustic phonetic and energy features," *Proc. ICSLP*, 2005.
[8] Chu, S.M., Marcheret, E., and Potamianos, G., "Automatic speech recognition and speech activity detection in the CHIL smart room," *Proc. MLMI*, pp. 332–343, 2005.
[9] Huang, J., Westphal, M., Chen, S., et al., "The IBM rich transcription spring 2006 speech-to-text system for lecture meetings," *Proc. MLMI* (same volume), 2006.
[10] Van Compernolle, D., "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," *Proc. ICASSP*, pp. 833–836, 1990.
[11] Armani, L., Matassoni, M., Omologo, M., and Svaizer, P., "Use of a CSP-based voice activity detector for distant-talking ASR," *Proc. Eurospeech*, pp. 501–504, 2003.
[12] Novak, M., Gopinath, R.A., and Sedivy, J., "Efficient hierarchical labeler algorithm for Gaussian likelihoods computation in resource constrained speech recognition systems," available on-line at: `http://www.research.ibm.com/people/r/rameshg/novak-icassp2002.ps`
[13] Gersho, A., and Gray, R.M., *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 3rd Ed., Ch. 11, 1993.
[14] Ramaswamy, G.N., Navratil, A., Chaudhari, U.V., and Zilca, R.D., "The IBM system for the NIST-2002 cellular speaker verification evaluation," *Proc. ICASSP*, vol. 2, pp. 61–64, 2003.

A Lightweight Speech Detection System for Perceptive Environments

Dominique Vaufreydaz, Rémi Emonet, and Patrick Reignier

PRIMA - INRIA Rhône-Alpes, Zirst, 655 avenue de d'Europe,
Montbonnot, 38334 Saint Ismier cedex, France
{Dominique.Vaufreydaz,
Remi.Emonet,Patrick.Reignier}@inrialpes.fr
http://www-prima.inrialpes.fr/

Abstract. In this paper, we address the problem of speech activity detection in multimodal perceptive environments. Such space may contain many different microphones (lapel, distant or table top). Thus, we need a generic speech activity detector in order to cope with different speech conditions (from close-talking to noisy distant speech). Moreover, as the number of microphones in the room can be high, we also need a very light system. The speech activity detector presented in this article works efficiently on dozens of microphones in parallel. We will see that even if its absolute score of the evaluation is not perfect (30% and 40% of error rate respectively on the two tasks), its accuracy is good enough in the context we are using it.

1 Introduction

The base principle of research in ubiquitous computing is to make the computer disappear from the human computer interfaces. Classical input and output devices (keyboard, mouse, screen, etc.) are replaced by other, less intrusive modalities such as voice recognition or computer vision. In order to conduct research in this domain, many research laboratories have equipped dedicated rooms with multiple sensors (cameras, microphones, etc.) and perceptual software (2D and 3D visual tracking systems, speech recognition systems, etc.). The goal is to enable the computer system to understand what the user is saying or doing. The computer system is then able to behave in accordance with the user's intentions. Such highly equipped spaces are often called perceptive environments.

In these environments, all audio sensors, from simplest ones to most complicated ones, have an important role to play. Speech is indeed one of the preferred and most natural communication channels in human to human interactions, and sounds are revealing of human activity. This is why many perceptual environments, such as in the CHIL project [1], are equipped with speech detection, speech recognition and acoustic localization systems. One requirement in such perceptive environments is to be able to process multiple and various microphones in parallel while fitting real time constraints.

Within the CHIL project [1], we are developing a speech detection system that fulfills the requirements of these perceptual environments. Although much research has already been conducted on this point and different approaches have been proposed

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 336 – 345, 2006.

(such as [2] and [2]), the problem is still open. In addition, we impose two constraints to what is done in most of other systems. Our system must be autonomous. It must be started and then run without human action. It must require neither training nor tuning each time the operating conditions change. We also want to keep our system as light as possible.

In section 2, we first give a description of our speech detection system. Section 3 then presents evaluations that were conducted and the results obtained in the NIST 06s evaluation[1]. Finally we will give a conclusion and some further works to be carried out in order to improve our system.

2 System Description

In our perceptive environment, the full SAD (Speech Activity Detection) system can run at the same time over one or many microphones. In this last case, there are two kinds of answer. First, a SAD decision is made at least for each microphone. We can also define a set of microphones in order to get an “ambient" SAD decision using for example multiple microphone arrays. A majority vote is done among all microphones to determine the current state. If speech and non speech votes are equal, the state of the global answer remains the same. This strategy is not optimal when using a large set of different microphones. The design of the system was made in order to run several systems in parallel over multiple groups of microphones.

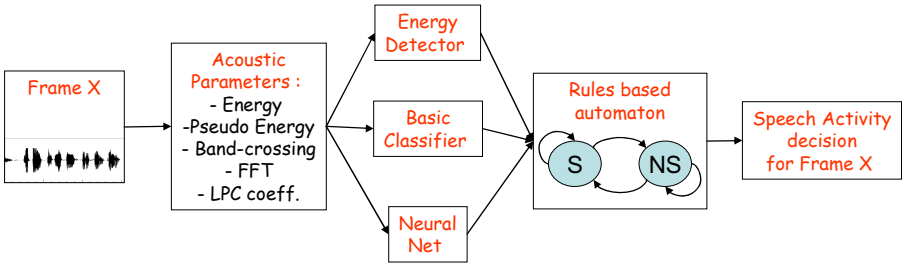


Fig. 1. Design of the SAD System

On each input, the current version of our SAD system works using several subsystems: an energy detector, a basic classifier and a neural net trained to recognize voiced segments like vowels for example. At each timestep, i.e. for each frame, each sub-system gives a speech or non-speech answer. Then a hand-made rule-based automaton determines the final result: whether or not there is speech activity. This tools is designed to be enhanced with complementary other subsystems.

2.1 Energy Detector

The energy detector uses pseudo energy (we do not sum square values of samples but only absolute values) to determine variation of the input signal energy. It works using

[1] See http://www.nist.gov/speech/tests/rt/rt2006/spring/

two couples of time delay/energy threshold: *TimeOn/EnergyOn* and *TimeOff/EnergyOff*. Simply speaking, if the pseudo energy goes over *EnergyOn* during *TimeOn*, the energy detector emits a *START_SPEAKING* event as a result. In the same way, if the pseudo energy falls under *EnergyOff* during *TimeOff*, the result is *STOP_SPEAKING*. For stable periods, the return values are respectively *STILL_SPEAK* and *NOT_SPEAK*.

As the system is designed to run permanently, it is obvious that these energy thresholds cannot always reflect the current voice/background noise energies. We added to the energy detector the ability to adapt dynamically these thresholds. A sliding window of 50 seconds of previous values for speech portion energy is maintained. In order to smooth threshold changes, the window is filled with the *EnergyOn* value at the initialization time. Then, when the global SAD state changes, for example when final answer of the SAD system goes from speech to non-speech, the training system computes the new threshold value. The system does exactly the same computation on a separate sliding window for *EnergyOff*.

In order to prevent usage of outliers in the online adaptation process, we do not use data from the beginning and the ending of speech or non-speech segments. We privilege inside segments which should be more stable. Thus, we eliminate *TimeOn* data from the beginning and *TimeOff* at the end of speech segments for our online adaptation. Identically, we do not use *TimeOff* and *TimeOn* data respectively from beginning and ending of non-speech segments. We also do not use data from segments that are not long enough. So, if a segment does not contain at least 1 second of interesting data, it is not use to compute new thresholds.

The final computation of new values for *TimeOn* and *TimeOff* do not use a simple average approach but a "median" one. We sort the adaptation data contained in the sliding windows and remove possible outliers, i.e. too low and too high values. At least, we keep only 60% of the data in order to re-estimate our thresholds. Our real adaptation time is thus 30 seconds (60% of 50s) for each threshold.

2.2 Basic Classifier

This classifier is dedicated to recognize and to tag specific sound classes: fricatives, low frequency sounds like computer or air conditioning fans, and other sounds. The first step is a hamming window and a Fast Fourier Transform (FFT) [4] to obtain the spectrum of the signal. The classifier deals with 5 identical sub-frequency bands from 1 to 8000 hertz where it computes energy. This classifier works only on signal recorded at 16000 hertz or higher sampling rates. In this last case, frequencies over 8000 hertz are not used.

With the 5 energy values, the module can classify the audio signal:

- if more than 90% of the total energy is concentrated in the 2 lowest bands, the sound is a *low frequency sound*
- if the energy in the 2 lowest bands is less than in all other bands, the sound is a *fricative*
- in all other cases, the sound remains *unclassified*.

2.3 Neural Network

The neural net is a multi-layer perceptron with 2 hidden layers. It uses as input coefficients computed on the input frames:

- Zero crossing: number of time the signal goes from a negative to a positive value and vice versa. Actually, we use a variant called band-crossing [5] that does not count oscillations in a band around 0.
- Energy: the sum of the square values of samples.
- 16 predictor coefficients: they are extracted from a speech analysis method called Linear Predictive Coding (LPC) [6]. We use the auto-correlation method combined with the Durbin recursion to compute them.

This module is the only sub-system that needs to be trained. The training was made once and for all on 1 hour of French speech extracted from the BREF corpus [7]. The phonetics labels used during the training phase are not the original BREF ones but were computed with RAPHAEL [8], a French recognizer. The training data were almost equilibrated, ~50% of female voice and ~50% of male voice. Result of this module can be *speech* or *non speech*.

2.4 Rules Based Automaton

This automaton is designed to integrate results from all subsystems to produce a final answer. It consists in 2 states (*speech* and *non speech*) with hand-made rules to change from one state to the other. The rules were defined using knowledge about each subsystem. In all cases, if all subsystems agree on the current state, i.e. when each result is a speech one[2], the system uses it. In the *speech* state, if the energy detector return value and at least one another result are *non speech*, we go into the *non speech* state. We do the same when the basic classifier returns *unclassified* and the neural net *non speech* or when the basic classifier gives *low frequency sound* as answer. Symmetrically, we defined the same type of rules for the *non speech* state.

3 Evaluation

3.1 Implementation

The implementation of the full system is made in C++ and can run on multiple operating systems. As explained above, the system can handle signal from 16 KHz to 44.1 KHz. During evaluation, the SAD system works on frames of 256 samples on a 16 KHz signal. Thus, the time precision of our system is 16 ms. Another important point: the system remains the same for all evaluation tasks. We do not have specific configuration or training for close talking or far field microphones.

3.2 RT06S Evaluation Data

The RT-06S evaluation is focused on the Meeting Domain interaction with two subdomains (or tasks). The first one consists of ten meetings recorded in a conference

[2] Speech events are *START_SPEAKING* or *STILL_SPEAK* for the energy detector, *fricative* for the basic classifier and *speech* for the neural net.

room in six different sites: it is called "*confmtg*". The second one, aka "*lectmtg*", is composed of several lectures with lecturer and question/answer speech. Each subdomain has different sensor setups, different levels of interactions and multiple structures of test excerpts. The reader may refer to [9] to find more detailed information about the evaluation data.

For each task, one may run its speech activity detection system in many different conditions. According to our system characteristics, we decided to evaluate our system on the following subset of conditions:

- Individual head microphone (*ihm*)
- Multiple Distant Microphones (*mdm*)
- Single Distant Microphone (*sdm*)
- All Distant Microphones (*adm*)
- Multiple Source Localization microphone Arrays (*msla*)

The final evaluation contains about 180 minutes of speech for the *confmtg* task and 145 minutes for *lectmtg*.

3.3 Evaluation Metrics

In this paper, we use two different metrics in order to test if our system fulfils our needs: a light and accurate system. To check if our SAD system is light enough, we compute the real-time factor as expressed in equation (1).

$$\text{Real - time factor} = \frac{\text{Total processing time}}{\text{Data time}} \tag{1}$$

This factor permits to know easily if the system can be real-time. If the real-time factor is less or equal to 1, so if the processing time is less or equal to the data time, the system can be considered as real-time. In the next section, we will check how many different inputs we can handle at the same time.

The other evaluation point concerns accuracy. We need to know how efficient our SAD system is. Within the RT06S evaluation, the SAD metric is time based [9]. The scoring system first computes the full speech time over the considered audio signal. Then using output from systems and reference files, manually annotated, we obtain the missed speech and the false alarms times. Doing that on all files from a condition for a given task and accumulating values, we can calculate the overall error rate on the given task using the equation (2):

$$\text{SAD Error} = \frac{\text{Missed speech} + \text{False Alarm time}}{\text{Speech time}} \tag{2}$$

In order to enrich this result, we built tools to compute some extra information. For each task and for each condition, we first extract for each talk, the SAD error rate. Then we decompose the Missed speech time in four time-weighted categories:

- *Full miss*: a complete speech event which is not detected;
- *Miss begin*: the beginning of a speech event is not detected;
- *Miss In*: middle part(s) of a speech event not tagged as speech;
- *Miss end*: the end of a speech event is not found.

Using this new information, we will be able to analyze more precisely the error committed by our SAD system.

3.4 Results

In this section, we detail the results of our system in the RT06s evaluation. First, we will introduce speed measurement and then, the accuracy of our SAD system.

3.4.1 Speed Evaluation

As we have already said, speed is a strong constraint for us. We work in interactive spaces and we need low latency application. It is not suitable for us to transmit a lot of data over the network. Thus, we need to be able to process a maximum of microphones on a single computer.

If we measure speed factor on a single 16 KHz audio signal, the computational time for 1 second of speech is 0.0076 second: in theory, the system can handle more than 130 channels at the same time. In practice, operating system scheduling, memory management and multi-channels SAD fusion can alter this result.

We built a test set using 300 seconds segments (containing voice) extracted from the RT06s evaluation database. The full SAD system ran over this set using firstly 1 segment, then 2, etc. Each time, segments were chosen randomly. We stop the test when the real-time constraint was violated, i.e. when the real-time factor goes over 1. The next figure shows the experimental results.

On this figure, one can found a dashed line obtained by linear regression on data. We also see a curve showing the real-time factor of our system regarding inputs.

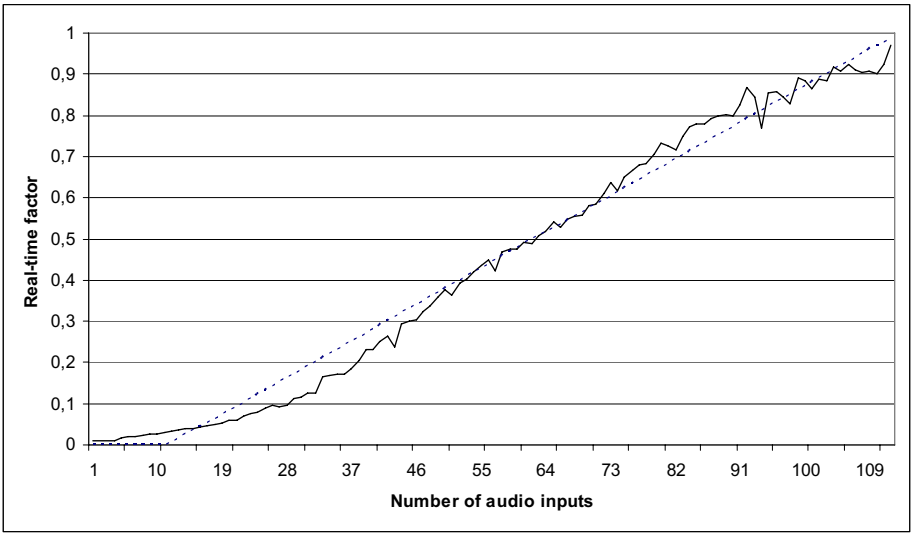


Fig. 2. Speed performance of our SAD system running on a single core unit of a processor and processing multiple inputs

We can first see that the real-time factor curve is not always increasing. This surprising phenomenon can be explained by the load of the computer and by the computation time that changes from one file to another (see 2.1). The other result which affords is that our system is linear-like over 50 inputs. It is very important because if we want to add some microphones, we can do it without rethinking the whole computer configuration.

Finally, our SAD system can process 112 streams which is a good score. In real conditions, i.e. when we do not process files, we must also consider that acquisition process will alter this result and certainly slow the system. Nevertheless, we can objectively say that we can process as many microphones as a sound card can record in real time.

3.4.2 Speech Activity Detection Accuracy

This section presents experimental results from the RT06s evaluation. In these experiments, starting energy thresholds of the energy detector were values empirically defined during previous research projects (NESPOLE! [10] and FAME [11]). The evaluation metrics of our system are given in the three following tables.

Table 1. Global results over the different tasks

Task	Condition	Overall Error Rate	Best Error Rate	Worst Error Rate
confmtg	ihm	78,54%	20,31%	1917,99%
	mdm	46,98%	13,12%	80,20%
	sdm	41,26%	18,48%	76,98%
lectmtg	adm	27,81%	3,73%	95,51%
	mdm	32,59%	4,29%	95,47%
	msla	30,61%	3,86%	100,00%
	sdm	33,87%	5,74%	85,29%

The first table above gives us general information about accuracy of our SAD system. The official results of the RT06s evaluation are given in the "Overall Error Rate" column. We can first see that our system is not accurate on the *confmtg-ihm* condition. For this condition, only speech coming from the main speaker must be tagged as speech. As our system is not design to do that (every speech segment can be tagged as speech, even if its energy is low), this result is not really significant: our worst score within this condition is 1918% of error. As we did not understand correctly this task before evaluating, we let the result in the previous table but we will not analyze more precisely this condition. Concerning others conditions of the *confmtg* task, we can see that our average error rate is ~43%. Our best scores, over one seminar, are not good (13% and 18%). We will check in the next section where our system fails. If we look to the *lectmtg* task, we can see that our global results are better: ~30% of error in average over all condition. Moreover, our best scores are good (from 3% to 6%) but our worst score stay high (up to 100%).

We will now trying to understand more precisely where the errors are. The following tables show our additional metrics computed first on the confmtg task, next on *lectmtg*.

Table 2. Detailed results of the *confmtg* task

Condition	Full Miss	Miss Begin	Miss End	Miss In	False Alarm	Error Rate
mdm	13,22%	3,31%	5,38%	24,97%	0,11%	46,98%
	(28,13%)	(7,04%)	(11,45%)	(53,15%)	(0,23%)	(100,00%)
sdm	1,63%	2,17%	4,39%	26,86%	6,22%	41,26%
	(3,95%)	(5,25%)	(10,63%)	(65,11%)	(15,07%)	(100,00%)

In this table, one can found the official evaluation error rate in the last column. The second sub-row of each entry gives the percentage of each type of error on the overall error rate.

On the multiple distant microphone (*mdm*) condition, our system is not good at all. 13% of the speech segments were entirely missed. 25% of the missed speech is within a speech turn. A good result is the false alarm error rate which is very low. This value is correlated to the 13% of full miss. Our SAD system did not make mistake on non speech segments but was insensible to some speech parts. Our starting thresholds were too high and not adapted to this condition. Moreover the system do not managed to adapt them online. Concerning the single distant microphone (sdm) condition, results are slightly better. In fact, we only have a *full miss* rate of ~2%. We see a rise of the *miss end* and *false alarm* rates. *Miss in* factor stay over 60% of our errors.

If we look globally at the previous table, we can say that ~60% of our errors are due to intra speech undetected portions. Even if we could artificially solve this problem by changing the *TimeOff* delay of our system, we do not want to do it. Field experiments have already shown at our laboratory that adapting system to an evaluation can lead to decrease drastically real world performances. Other metrics can not be averaged because there are too distant between the two conditions.

The table below introduces more accurate results achieved on the *lectmtg* task.

Table 3. Detailed results of the *lectmtg* task

Condition	Full Miss	Miss Begin	Miss End	Miss In	False Alarm	Error Rate
adm	3,65%	3,06%	4,25%	12,20%	4,66%	27,81%
	(13,12%)	(10,99%)	(15,28%)	(43,87%)	(16,75%)	(100,00%)
mdm	5,83%	3,52%	5,00%	11,59%	6,65%	32,59%
	(17,89%)	(10,81%)	(15,34%)	(35,56%)	(20,40%)	(100,00%)
msla	4,52%	3,65%	4,69%	12,51%	5,24%	30,61%
	(14,75%)	(11,92%)	(15,33%)	(40,88%)	(17,12%)	(100,00%)
sdm	3,73%	4,16%	6,11%	13,93%	5,94%	33,87%
	(11,00%)	(12,29%)	(18,05%)	(41,13%)	(17,53%)	(100,00%)

The results of our SAD system over the *lectmtg* task are better than on the *confmtg*. In absolute, the overall error rate is 10% lower (~30% overall error rate). We can also remark that the percentages for all conditions are similar: in average 4.5% of *full miss*, 3.6% of *miss begin*, 5% of *miss end*, 12.5% of *miss in* and 5.6% of *false alarm*[3]. We can quickly see that the *miss in* errors represents a huge amount of the error rate

[3] As a comparison point, the false alarm rate for a system always answering *speech* is 25.44% on this task.

(40%). *False alarm* and *miss end* follow with almost 20% of the errors. We can analyze these results saying that the data of the *lectmtg* evaluation are closer to the capabilities of our system than *confmtg*. We still see a huge amount of boundaries problems, lost intra speech segments remain our major problem. Finally, the good outcome is that our system can detect ~95% of the speech turns even if the boundaries are not precisely located.

At end, we can draw some conclusions on this evaluation. Our SAD system is not perfect if we look only at the final percentage. Most of the time, problems are boundaries (*miss begin*, *miss in*, *miss end*) and a non negligible part is *false alarm*. After looking at some labels and reference files, we can say that these problems seem to be less present at the end of the seminars. The adaptation process seems to refine the thresholds but with a too long latency. As our system is designed to run permanently, it is usually not a problem. During evaluation, the SAD system starts from scratch for each file. As we already said, we need transitions in order to compute new values. If we do not have enough transitions, it is obvious that we will not have a suitable adaptation process. For the next evaluation, we should consider to use our system in real condition but for this experiment, we decided that grouping seminar by collecting site, thus by similar recording equipments, settings and conditions, can be considered as cheating. We have chosen not to do so.

For us, the major result of our system for the RT06s evaluation is the low percentage of *full miss* speech turns (13% of the *mdm* and 1.6% for the *sdm* condition of the *confmtg* task and <5% in average for the *lectmtg* task). This score validates the usability of our SAD system in the context we are using it: detecting speech turns and people interactions for context modeling.

4 Conclusion and Future Work

Our speech detection system exposes performances that make it suitable for our projects and goals such as CHIL [11] or [12]. Actually, we do not want to use it for automatic speech recognition or diarization but for interaction and context modeling. Thus, we do not need precise speech boundaries but only to be sure to detect interaction between people so at least a part of each speech turn. If we look to the previous section, we can see that this goal is fulfilled: the *full miss* score is low. We think that this result is good for a generic system not trained on twin data of the evaluation that can run in different working environment (smart office, conference room, amphitheater, etc.) without any preliminary training/adaptation. Concerning our speed constraint, we saw that our system is successful because we can process many inputs in parallel. Following proposed improvements will be integrated carefully in order to preserve this capability.

For future work, experiments described in this paper have shown that some improvements of our system still need to be carried out. First of all, the online adaptation of the energy thresholds has to be refined. It could improve the performance of the system but decrease the system accuracy on adaptation failure. Next, we also envision extending the training of our neural network. We still want to keep an *a priori* training for this neural network while including different kind of speech recorded in different environment and different conditions (lapel and distant microphone). We

think that doing this will provide us with a more generic speech detection system. We are currently preparing the corpus required by such extended learning.

We also want to improve our fusion scheme. We can substitute our rule based approach by a Bayesian one. The current expert-defined rules are rigid and could be advantageously replaced by a Bayesian fusion process. Moreover, if we plan to add other speech activity detection sub-systems, it will be difficult to rebuild a new set of rules. Doing this Bayesian training will also require having a learning corpus.

The last source of improvement would be to take advantage of all the available data in our perceptive environment. Our system could use visual information (facial features, visual clues of sound events, etc.) to improve the performance of speech detection.

References

1. D. Macho, J. Padrell, A. Abad, C. Nadeu, J. Hernando, J. McDonough, M. Wolfel, U. Klee, M. Omologo, A. Brutti, P. Svaizer, G. Potamianos, S.M. Chu. Automatic Speech Activity Detection, Source Localization, and Speech Recognition on the Chil Seminar Corpus. In IEEE International Conference on Multimedia & Expo, January 2005.
2. J. Ramirez, J. Segura, C. Benitez, A. de la Torre, A. Rubio. Efficient voice activity detection algorithms using long-term speech information. Eurospeech'97, pages, 1997.
3. A. Martin, D. Charlet, L. Mauuary. Robust Speech/Non-Speech Detection Using LDA Applied to MFCC. In Proc. ICASSP, vol. 1, 237-240, Salt Lake City, May 2001.
4. M. Frigo, and S.G. Johnson, The Design and Implementation of FFTW3, special issue on "Program Generation, Optimization, and Platform Adaptation", volume 95, pages 216-231, 2005.
5. J. Taboada, S. Feijoo, R. Balsa, C. Hernandez, Explicit estimation of speech boundaries, IEEE Proc. Sci. Meas. Technol., vol. 141, pp. 153-159, 1994
6. L. Rabiner, B.H. Juang, Fundamentals of Speech Recognition, Prentice Hall PTR, ISBN 0-130-15157-2, 1993.
7. L. Lamel, J.L. Gauvain, M. Eskenazi, BREF, a large vocabulary spoken corpus for French. In Proc Eurospeech'91, Genova (Italia), 1991.
8. D. Vaufreydaz, Modélisation statistique du langage à partir d'Internet pour la reconnaissance automatique de la parole continue, Ph.D. in Computer Science at Joseph Fourier University, Grenoble (France), 226 pages, January 2002
9. Spring 2006 (RT-06S) Rich Transcription Meeting Recognition Evaluation Plan, http://www.nist.gov/speech/tests/rt/rt2006/spring/docs/rt06s-meeting-eval-plan-V2.pdf.
10. F. Metze, J. Mc Donough, H. Soltau, A. Waibel, A. Lavie, S. Burger, C. Langley, L. Levin, T. Schultz, F. Pianesi, R. Cattoni, G. Lazzari, N. Mana, E. Pianta, L. Besacier, H. Blanchon, D. Vaufreydaz, L. Taddei, The Nespole! Speech-to-Speech Translation System, Human Language Technologies 2002, San Diego - California (USA), 6 pages, mars 2002.
11. F. Metze, P. Gieselmann, H. Holzapfel, T. Kluge, I. Rogina, A. Waibel, M. Wolfel J. Crowley, P. Reignier, D. Vaufreydaz F. Bérard, B. Cohen, J. Coutaz, S. Rouillard V. Arranz, M. Bertran,, H. Rodriguez, The "FAME" Interactive Space, 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Edinburgh - UK, 4 pages, February 2005.
12. O. Brdiczka, J. Maisonnasse, P. Reignier, Automatic Detection of Interaction Groups. In Proc. Int'l Conf. Multimodal Interfaces, October 2005.

Robust Speaker Diarization for Meetings: ICSI RT06S Meetings Evaluation System

Xavier Anguera[1,2], Chuck Wooters[1], and Jose M. Pardo[1,3]

[1] International Computer Science Institute, Berkeley CA 94704, USA
[2] Technical University of Catalonia, Barcelona, Spain
[3] Universidad Politecnica de Madrid, Madrid, Spain
{xanguera,wooters,jpardo}@icsi.berkeley.edu

Abstract. In this paper we present the ICSI speaker diarization system submitted for the NIST Rich Transcription evaluation (RT06s) [1] conducted on the meetings environment. The presented system is based on the RT05s system, which uses agglomerative clustering with a modified Bayesian Information Criterion (BIC) measure to decide which pairs of clusters to merge and to determine when to stop merging clusters. In this year's system we have eliminated any remaining need for training data, therefore increasing robustness. In our primary system we have introduced several improvements from last year. First, we use a new training-free speech/non-speech detection algorithm. Second, we introduce a new algorithm for system initialization. The third improvement is the use of a frame purification algorithm to increase cluster discriminability. Finally, we describe the use of inter-channel delays as features. We explain each of these improvements and show our system's results on the official evaluation data using hand-aligned references and forced-alignments. We also analyze some of the results and propose improvements.

1 Introduction

The goal of a diarization system is to locate homogeneous regions within an audio segment and consistently label them for speaker, gender, music, noise, silence, etc. Within the framework of the Rich Transcription 2006 Spring Meeting Recognition Evaluation, the labels of interest were solely speaker and silence regions. This year's evaluation continues to focus on two meeting subdomains: the conference room, as in the RT04s and RT02s evaluations, and the lecture room, with seminar-like meetings. In each subdomain, a test set of about two hours was distributed. Participant's systems were asked to answer the question "Who spoke when?". The systems were not required to identify the actual speakers by name, but just to consistently label segments of speech from the same speaker. Prior art in this task can be seen in the different systems participating at RT05s [2], [3], [4]. Performance was measured based on the percentage of audio that was incorrectly assigned. This year was our second participation in the speaker diarization task. The speaker diarization system we used is based on last year's system (see [5]). Our system is based

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 346–358, 2006.

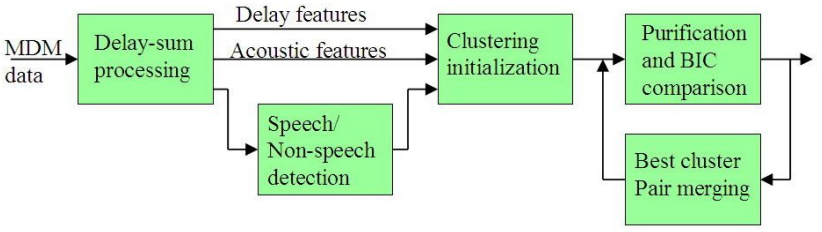


Fig. 1. RT06s Speaker Diarization system blocks diagram

on an agglomerative clustering system developed by Ajmera et al. (see [6]). Its primary advantage is that it requires no pre-trained acoustic models and therefore is robust and easily portable to new tasks.

Some of the improved algorithms are: A new hybrid speech/non-speech detector which combines an energy-based detector with a model based decoder back-to-back in order to avoid the need for outside training data. Also, a new system initialization and an automatic technique for selecting the number of initial clusters. We have also introduced an improved delay&sum algorithm to enhance the signal when multiple acoustic channels are available and a new frame-based purification algorithm that replaces last year's segment-based algorithm and enhances cluster discriminability. Finally, the use of inter-channel time differences as an extra feature stream for the diarization system.

In next section we review the general blocks on which the MDM system is based, sections 3 through 7 introduce the main changes in the system from the last submission in RT05s. Section 8 introduces the use of forced-alignments for this year's development and section 9 presents the main characteristics of the systems submitted. Finally, section 10 shows the systems results and 12 draws some conclusions.

2 Speaker Diarization System

As explained in [5], our speaker diarization system is based on an agglomerative clustering technique. Its main blocks are shown in figure 1 for the case of multiple microphones. It initially splits the data into K clusters (where K must be greater than the number of speakers and is chosen using the algorithm presented in [7]), and then iteratively merges the clusters (according to a metric based on ΔBIC) until a stopping criterion is met. Our clustering algorithm models the acoustic data using an ergodic hidden Markov model (HMM), where the initial number of states is equal to the initial number of clusters (K). Upon completion of the algorithm's execution, each remaining state is taken to represent a different speaker. Each state in the HMM contains a set of MD sub-states, imposing a minimum duration on the model (we use $MD \simeq 3$ seconds). Within the state, each one of the sub-states shares a probability density function (PDF) modelled via a Gaussian mixture model (GMM) for each particular data-stream.

The system works as follows:

1. If more than one recorded channel is available for a given meeting recording, combine them all into a single "enhanced" channel using a delay&sum algorithm further described in [8].
2. Run speech/non-speech detection on the "enhanced" data using the speech/non-speech algorithm presented in [9] and explained in section 3.
3. Extract acoustic and delay features from the data and remove non-speech frames from the agglomerative processing.
4. Estimate the number of initial clusters K using the algorithm presented in [7].
5. Create models for the K initial clusters using the new cluster initialization algorithm explained in section 4 and in [10].
   (a) Run a Viterbi decode to resegment the data.
   (b) Retrain the models using the Expectation-Maximization (EM) algorithm and the segmentation from step (a). Iterate between (a) and (b) until the segmentation stabilizes.
   (c) Select the cluster pair with the largest merge score (based on ΔBIC) that is > 0.0 using the frame purification technique introduced in [11] and section 5.
   (d) If no such pair of clusters is found, stop and output the current clustering.
   (e) Merge the pair of clusters found in step (c). The models for the individual clusters in the pair are replaced by a single, combined model.
   (f) Go to step (a).

As the stopping criterion for clustering and a distance measure for merging, we use a variation of the commonly-used BIC [12]. The variation that we use was introduced by Ajmera et al. [6], and consists of the elimination of the tunable parameter λ by ensuring that, for any given ΔBIC comparison, the difference between the number of free parameters in both models is zero.

One of the main overall changes for this year is that we eliminated all remaining dependency of our system on training data. This was achieved by the creation of a training-free speech/non-speech detector introduced in the next section. Furthermore, this year we introduce the use of data other than acoustic data for clustering by successfully using the delays between channels (in the MDM condition) as a new feature stream in the agglomerative clustering. This is further explained in section 6. Apart from these, a new clustering initialization algorithm and a frame purification algorithm contributed to the increase in the system's robustness and therefore improved its performance. Last year's segment purification algorithm was not used this year. The following sections introduce all these techniques.

3 Speech/Non-speech Detection Algorithm

In speaker diarization it is important to use a speech/non-speech detector as non-speech frames adversely affect the clustering performance. In the RT05s

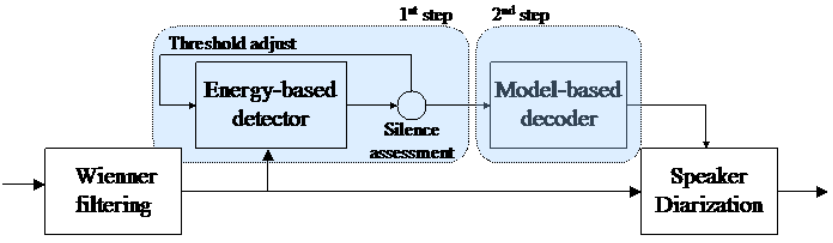


Fig. 2. Speech/non-speech block diagram

evaluation the speech/non-speech system we were using was based on pre-trained acoustic models for both speech and non-speech. This forced the readjustment of the models every time a new environment was to be processed, e.g. "conference room" versus "lecture room" data. For this year's evaluation we have developed a new speech/non-speech detector [9] that is train-free and therefore more robust to unseen data, as long as the main non-speech event in the recording is silence (which is a common trait of meeting data).

The system shown in figure 2 is a hybrid energy-based detector and model-based decoder. In the first stage, an energy-based detector finds all segments with low energy, while applying a minimum segment duration. An energy threshold is set automatically to obtain enough non-speech segments. In the first pass it takes a very low value and it increases incrementally while the number of non-speech frames falls under 100 and bigger than 10 (chosen empirically). At that point the segmentation is used to train speech and non-speech models in the second module and then several iterations of Viterbi segmentation and model retraining take place, finally outputting the speech/non-speech segmentation when the likelihood converges. In the system we need to define three parameters: the minimum durations for speech/non-speech in the energy module, minimum duration for speech/non-speech in the cluster module and the number of components used to model speech and non-speech in the cluster module. The parameters were tuned using the forced-alignment segmentations on the development set. As shown in [9] even though the miss and false alarm errors are equivalent to those obtained using the pre-trained system, the new system is more robust to changes in the data and appears to be a better fit with the following diarization module.

4 Cluster Initialization Algorithm

In order for the agglomerative clustering to work properly in obtaining the optimum number of speakers for a particular recording, we need to initialize the system with K (where $K > K_{true}$ the true number of speakers) clusters containing acoustically homogeneous data.

Past experiments, using k-means initialization and other techniques, have indicated that one very good option was to do a linear initialization of the data, where K clusters are generated by evenly splitting the acoustic data and then

performing several iterations of model training and resegmentation to allow for homogeneous acoustic data to come together. Although a very simple technique that works extremely well for some cases, in many other cases the resulting clusters contain more than one speaker which affects the (5c-d above) stopping criterion causing the final DER to increase.

The new initialization algorithm, explained in [10], consists of three stages of processing. First, speaker change detection using the Bayesian Information Criterion (BIC) metric (modified not to use a penalty term, as in out clustering system) is used to define acoustically similar segments by finding speaker change points via a scrolling window composed of two one-second regions. The second stage performs a bottom-up clustering by iteratively choosing speaker segments close to an initial segment (friends) to form one cluster and then selecting the segment most dissimilar to all existent clusters (enemy) to initialize the next cluster. Once K clusters are defined, their models are created and a segmentation is performed to assign all remaining segments to either model (third step). Using this technique, we obtain an increase in cluster purity right after the initialization process and a general improvement of the overall DER.

5 Frame Purification for Cluster Comparison

By using an agglomerative clustering technique, the system's performance heavily relies on the metric used to compare the similarity between cluster pairs as well as the clustering stopping criterion. Non-speech data is one of the main causal factors of anomalous behavior, which is one of the reasons a speech/non-speech detector is being used prior to the clustering process. After filtering the non-speech, the data considered to be speech still contains small non-speech segments (normally silence segments in the meeting environment) and other unvoiced speech which affects the cluster's modelling and degrades discriminability between clusters.

The frame purification algorithm (explained in [11]) detects and prevents such acoustic frames from affecting the models during the BIC comparison. To do so, it uses a metric related to the likelihood of the frames given the acoustic model. It is shown that when the cluster model's complexity is greater than two gaussian mixtures, most non-speech frames obtain the highest likelihoods, indicating that these are modelled with a narrower variance. A nice improvement in the model's discriminability is obtained by removing all frames with scores in the top 20% of the likelihood when training models for BIC comparison. This method is demonstrated to work better than filtering based on average frame energy [11].

6 Use of Inter-channel Delays in Clustering

Possibly this year's most effective improvement is the inclusion of inter-channel delays for the tasks where more than one microphone is available (see [13]). The delays are a byproduct of the delay&sum processing. For inclusion in the clustering, the delays are computed between a reference channel and all other

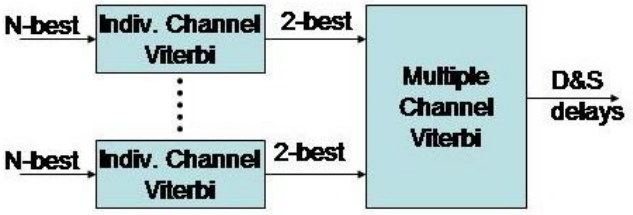


Fig. 3. Delay and Sum double-viterbi delays selection

channels at the same rate as the acoustic features and then post-processed in the same way as in the delay&sum presented below. The delays are initially modelled using single gaussian mixtures, with the same minimum duration as the acoustic features and share the speaker segmentation with the acoustic models. When two clusters are set to be merged their delay models are combined in the same way as the acoustic models.

Both the delay models and the acoustic models are used to classify the data into the different clusters via a Viterbi segmentation and for cluster comparison using BIC. We make the assumption that delay and acoustic information is uncorrelated and therefore can be modelled with separate models. The joint log-likelihood for any given frame is computed as:

$$lkld(x_{aco}[n], x_{del}[n]|\Theta_{aco}, \Theta_{del}) = \alpha \cdot lkld(x_{aco}[n]|\Theta_{aco}) + (1-\alpha) \cdot lkld(x_{del}[n]|\Theta_{del}) \quad (1)$$

Where Θ_{aco} is the acoustic model, Θ_{del} is the delay model and α weights the effect of each model in the system. The value for α needs to be tuned using development data. In our work, we found that a good value for $\alpha \sim 0.9$.

7 Delay&Sum Improvements

Whenever more than one channel is available for processing, a delay&sum beamforming is applied in order to obtain one single "enhanced" channel. The system used is based on last year's (see [5]), with four added improvements. The first improvement affects the noise filtering. Last year's submission filtered out any delay with cross-correlation value smaller than 0.1 since very low signal correlations indicate less reliability. This caused noisy meetings (or recordings with the lowest quality microphones) to have more frames filtered than in "cleaner" meetings. This year's submission computes a global histogram of all delays, in all channels, and determines the threshold at 10%. As in last year's system, any frame labelled as noisy is replaced by the delay from the previous usable frame, ensuring continuity of the delays.

Another improvement this year involves the delays selection among the N-best GCC-PHAT. As seen in figure 3 we apply a 2-level Viterbi decoding. The first

level consists of a local individual-channel decoding where the 2-best delays are chosen from the N-best delays computed for that channel at every frame. Each possible state has an emission probability equal to the GCC-PHAT value for each delay, and the transition probability between two nodes is inversely proportional to the distance between its delays, ensuring that the N-best probabilities in a particular instant sum up to 1. The second-level viterbi decoding finds the best possible path given all combinations of delays from the 2-best delays in each channel. The emission probabilities are the product of the individual GCC-PHAT values of each considered delay, and the transition probabilities are computed as in the first step, summing all delays distances from all considered delays, and normalized to sum to 1. In both cases the transition probabilities are weighted to emphasize its effect in the decision of the best path (we use a weight equal to 25 in both steps). This newly-introduced technique aims at finding the optimum tradeoff between reliability (cross-correlation) and stability (distance between contiguous delays). We value the second the most as our aim is to obtain an improved signal, avoiding quick changes of the beamforming between acoustic events.

The other two improvements affect the way that channels are summed after their relative delays are obtained. One of last year's post-eval improvements included an adaptive weighting for each individual channel (see [5]). This year we enhanced this concept by using the average cross-correlation between all channels (given the selected delays) to find the relative weights between the channels at each point. This value is also used to eliminate summing any channels with a relative weight smaller than $\frac{1}{4N}$ where N is the number of channels.

The delay&sum beamforming is used to enhance the signals to be used in this year's Speaker diarization systems as well as in the automatic speech recognition (ASR) submissions [14] for both conference and lecture tasks.

8 Use of Forced-Alignments

During this year's development period we experienced difficulties when using hand-made reference files, mostly when scoring on speaker overlap regions. By comparing the hand-made references with the acoustic data we observed that varying amounts of extra padding were inserted around each speaker overlap region, making its duration much longer than the actual acoustic event. We also observed some speaker overlap regions not labelled as such and some speaker overlap labels on non-speaker-overlap regions (although some speaker overlap might be noticed on the IHM channels, its volume is too low to be perceived in the MDM channels). All these artifacts create an extra amount of missed-speech error and of speaker error, which is not consistent over the different evaluation datasets (possibly as the transcription team changes their transcription guidelines). In general, we believe that the hand-made speaker segmentation references show too much transcriber dependency to be able to compare results from different years or to create a consistent and robust speaker diarization system.

For this year's system development we have taken the initiative to use references derives from forced-alignments. We generated the forced-alignments from the hand-transcribed spoken text with the individual IHM acoustic data. This was done at ICSI using the ICSI-SRI speech-to-text system presented for the RT05s evaluation ([15]). The use of forced aligned references was initially proposed by NIST for this year's meetings evaluation, although it was finally not applied.

In table 1 we compare the results using the same system output (a similar version to this year's primary MDM system) evaluated using either hand-aligned references or forced-alignments. We observe a change of between 2% and 5% in DER from non-overlap to overlap speech in the forced-alignment results, while there is a change from 6% to 15% in the hand-alignments, indicating the higher variability in the transcription of speaker overlaps. Additionally, in the evaluations up to RT05s, the non-overlap results are very similar between the hand-aligned and the FA, but in RT06s the difference is very large.

Table 1. Comparison of the DER for all meetings evaluation campaigns using hand-alignments or forced-alignments

Evaluation	MDM Hand-align		MDM Force-align	
campaign	non ovl.	ovl.	non ovl.	ovl.
RT02s	20.79%	26.95%	19.93%	21.89%
RT04s	15.44%	30.55%	13.98%	17.01%
RT05s	10.41%	18.73%	12.52%	15.06%
RT06s	23.06%	36.99%	16.46%	21.19%

Due to the fact that we performed our development experiments using force-aligned references while the eval was scored using hand-alignments, we observed a large increase in our missed-speech error. In most cases this is due to the difference in the extra padding applied to the speaker overlap regions and to the difference in the non-speech labelling criteria (the rule of 0.3sec minimum is applied to the forced-alignments).

9 Evaluation System Descriptions

This year we presented a total of 23 systems in the multiple tasks and subtasks of the evaluation. Each system uses one or more of the improvements presented above. Across tasks, systems with the same ID are equal or very similar, just differing on a few parameters. Their characteristics are:

p-wdels: This is the primary system presented this year for all multi-microphone conditions. It uses all proposed techniques in this paper, and all changes in the diarization code from last year's evaluation.

c-newspnspdelay: This system is presented for the multi-microphone cases and is composed of RT05s evaluation code using this year's delay&sum algorithm, this year's hybrid speech/non-speech detector and taking advantage of the delays for clustering. It uses a minimum duration of 3 seconds, 1/5 initial gaussian mixtures for delays/acoustics and a split weight of 0.1/0.9 between the streams. It is intended to measure the improvements of using the delay features and the new speech/non-speech detector.

c-wdelsfix: This system is identical to p-wdels in all parts except the decision of the initial number of clusters, which is fixed to 16 and 10 clusters for conference and lecture rooms, respectively. It intends to compare the robustness of the initial number of clusters selection.

c/p-nodels: This system contains all of this year's improvements with respect to delay&sum (when available, in MDM), speech/non-speech detection and other diarization algorithms except the inclusion of the delays as an extra feature stream.

c-oldbase: This system uses all improvements in delay&sum (when available, in MDM) and speech/non-speech detection while using the RT05s core speaker diarization system. It is meant to serve as a baseline result for systems this year.

c-guessone: This system guesses one speaker for the entire show. In RT05s we presented this system as our primary system for lecture room data. Since the lecture room data is primarily composed of a single speaker, we believe that this is a reasonable baseline. This year we again present this system as a baseline lecture-room system to be compared with our other lecture-room systems.

10 Evaluation Results

In this section we present the scores for all of the ICSI systems presented in the RT06s evaluation in the speaker diarization (SPKR) task and the speech activity detection (SAD) task. In tables 2 and 3, we show the SPKR results both for conference and lecture room data, and in table 4 we show results for SAD. In all cases we use both the official hand-made references and the forced-alignment references computed as explained previously. In general this year's results using hand-alignments are much worse than in previous years for conference room, which is not so pronounced when evaluating using the forced alignments. This might be due to the increased complexity of the data and of a decrease in the quality of the hand-generated transcriptions for this year's evaluation.

In the SPKR task for conference room a substantial improvement can be seen between the first three systems in MDM and the last two due to using delays as features in diarization. In lecture room data (Table 3, third column) the use of

Table 2. Results for Speaker Diarization, conference room environment

Cond.	System ID	%DER MAN	%DER FA
MDM	p-wdels	35.77%	19.16%
	c-newspnspdelay	35.77%	20.03%
	c-wdelsfix	38.26%	23.32%
	c-nodels	41.93%	27.46%
	c-oldbase	42.36%	27.01%
SDM	p-nodels	43.59%	28.25%
	c-oldbase	43.93%	28.21%

Table 3. Results for Speaker Diarization, lecture room environment

Cond.	System ID	%DER MAN	%DER MAN(subset)	%DER FA(subset)
ADM	p-wdels	12.36%	11.54%	10.56%
	c-nodels	10.43%	10.60%	9.71%
	c-wdelsfix	11.96%	12.73%	11.58%
	c-guessone	25.96%	23.36%	24.51%
MDM	p-wdels	13.71%	11.63%	10.97%
	c-nodels	12.97%	13.80%	13.09%
	c-wdelsfix	12.75%	12.95%	12.34%
	c-guessone	25.96%	23.36%	24.51%
SDM	p-nodels	13.06%	12.47%	11.69%
	c-guessone	25.96%	23.36%	24.51%
MSLA	p-guessone	25.96%	23.36%	24.51%

delays affects negatively the performance, possibly due to talkers moving around the room (delays argue for a different speaker for each location).

In general the more microphones available for processing, the better the results. As the diarization system is the same, the improvement is due to the delay&sum processing. This is clear in the conference room data, while in the lecture room data, the results are mixed. We believe this is due to the difference in quality between the microphone used in SDM and all others.

In the lecture room results shown in Table 3 we compare the manual and forced-alignment DER for all systems submitted. The third column shows the results using the latest release of the manual reference segmentations (18 meeting segments). When generating the forced-alignments using the IHM channels from each individual speaker we could not produce them for the meeting segments containing speakers not wearing any headset microphone. The last column shows results using forced-alignment references for a subset of 17 meeting segments containing all speakers who wore a headset microphone. The second to last column shows results using this same subset and using hand-alignments for comparison purposes.

Results using FA references are much better than using hand-alignments in the conference room, while they remain similar in lecture room (with a constant improvement of 0.5% to 1% for FA). We believe the conference room manual

Table 4. Results for Speech Activity Detection (SAD). Results with * are only for a subset of segments.

Env.	Cond.	%DER MAN (%MISS, %FA)	%DER MAN(subset)	%DER FA
Conference	MDM	23.51 (22.76, 0.8)	–	11.10 (7.80, 3.30)
	SDM	24.95 (24.24, 0.8)	–	11.50 (8.80, 2.70)
Lecture	ADM	13.22 (9.3, 3.9)	7.9* (5.0, 2.9)	7.2* (3.7, 3.5)
	MDM	13.83 (9.3, 4.5)	6.5* (5.0, 1.5)	5.6* (3.6, 2.0)
	SDM	14.59 (10.0, 4.6)	7.2* (4.5, 2.7)	6.7* (3.3, 3.4)

references still contain many problems, which have been filtered out in the lecture room references after several redistributions of references.

In table 4 we show the results of our systems on conference and lecture room data for the SAD task, using the new speech/non-speech detector developed for this year's evaluation.

This year's speech/non-speech detector was developed using forced-alignment (FA) data. Therefore the results of the SAD are better as shown in the forced-alignment column. The increase in % MISS in the hand-aligned conference data is probably due to silence regions (greater than 0.3s) that are correctly labelled by the FA transcriptions but are considered speech by the hand-alignments.

As we did for the diarization experiments, we created a subset of meetings to appropriately evaluate the lecture room systems using forced-alignment references, and the counterpart hand-alignments for completeness. One initial observation is that the error rate decreases dramatically when evaluating only a subset of the shows using hand-alignments. a possible explanation for this is transcription errors produced due to the lower quality of the non-headset microphones used in the eliminated set of meetings.

As in the diarization results, in these experiments we also obtain better results with more microphones. When comparing the forced-alignment with the hand-alignment subset, the first group keeps a better balance between misses and false alarms, indicating that parameters defined in development translate robustly to the evaluation data.

Overall, we see an improvement this year with the use of delays between microphones as a feature in the diarization process for conference room data, while mixed results are obtained in lecture room. Also, a general improvement is observed using delay&sum on as many microphone signals as possible.

11 Overlapping Speaker Detection

Given that this year's evaluation counts speaker overlap errors in the main metric, we initially spent some time trying to build an overlap detector. In all our experiments we managed to lower the missed speaker error but at the cost of increasing the overall diarization error. We stopped research in this area when we started developing our system using forced-alignments, as the speaker overlap error in this case is less than 5%.

We performed experiments both in the diarization module and in the beamforming module. In diarization we tried a final decoding pass using the resulting speaker models and also all combinations of speaker pairs in order to detect speaker overlap. In the beamforming module we tested several metrics comparing the N-best cross-correlation values under the assumption that two speakers get consistently two main peaks in the correlation function.

12 Conclusions

This paper presents ICSI's submissions to the RT06s speaker diarization and SAD evaluation campaigns. This year's system contains four major improvements from last year. They are: a new training-free speech/non-speech detector, a new initialization algorithm, an improved cluster comparison algorithm, and the use of inter-channel delays as features in the diarization process. In this paper we review the basic system operation and we describe each of the improvements. Results are shown for the submitted systems while comparing the suitability of using hand-alignments versus forced-alignment references. Finally we describe some experiments in detecting speaker overlap.

References

1. NIST rich transcription evaluations, website: http://www.nist.gov/speech/tests/rt.
2. D. Istrate, C. Fredouille, S. Meignier, L. Besacier, and J.-F. Bonastre, "NIST RT05S evaluation: Pre-processing techniques and speaker diarization on multiple microphone meetings," in *NIST 2005 Spring Rich Transcrition Evaluation Workshop*, Edinburgh, UK, July 2005.
3. S. Cassidy, "The macquarie speaker diarization system for RT05S," in *NIST 2005 Spring Rich Transcrition Evaluation Workshop*, Edinburgh, UK, July 2005.
4. D. van Leeuwen, "The TNO speaker diarization system system for NIST RT05s for meeting data," in *NIST 2005 Spring Rich Transcrition Evaluation Workshop*, Edinburgh, UK, July 2005.
5. X. Anguera, C. Wooters, B. Peskin, and M. Aguilo, "Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system," in *RT05s Meetings Recognition Evaluation*, Edinburgh, Great Brittain, July 2005.
6. J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *ASRU'03*, US Virgin Islands, USA, Dec. 2003.
7. X. Anguera, C. Wooters, and J. Hernando, "Automatic cluster complexity and quantity selection: Towards robust speaker diarization," in *MLMI'06*, Washington DC, USA, May 2006.
8. ——, "Speaker diarization for multi-party meetings using acoustic fusion," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Puerto Rico, USA, November 2005.
9. X. Anguera, M. Aguilo, C. Wooters, C. Nadeu, and J. Hernando, "Hybrid speech/non-speech detector applied to speaker diarization of meetings," in *Speaker Odyssey 06*, Puerto Rico, USA, June 2006.
10. X. Anguera, C. Wooters, and J. Hernando, "Friends and enemies: A novel initialization for speaker diarization," in *Proc. ICSLP*, Pittsburgh, USA (to appear), September 2006.

11. ——, "Purity algorithms for speaker diarization of meetings data," in *Proc. ICASSP*, Toulouse, France, May 2006.
12. S. Shaobing Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Virginia, USA, Feb. 1998.
13. J. M. Pardo, X. Anguera, and C. Wooters, "Speaker diarization for multiple distant microphone meetings: Mixing acoustic features and inter-channel time differences," in *Proc. ICSLP*, September 2006.
14. A. Janin, A. Stolcke, X. Anguera, K. Boakye, O. Cetin, J. Frankel, and J. Zheng, "The ICSI-SRI spring 2006 meeting recognition system," in *Proceedings of the Rich Transcription 2006 Spring Meeting Recognition Evaluation*, Washington, USA, May 2006.
15. A. Stolcke, X. Anguera, K. Boakye, O. Cetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng, "Further progress in meeting recognition: The icsi-sri spring 2005 speech-to-text evaluation system," in *RT05s Meetings Recognition Evaluation*, Edinburgh, Great Brittain, July 2005.

Technical Improvements of the E-HMM Based Speaker Diarization System for Meeting Records

Corinne Fredouille and Grégory Senay

LIA-University of Avignon - BP1228 - 84911 Avignon Cedex 9 - France
{corinne.fredouille,gregory.senay}@lia.univ-avignon.fr

Abstract. This paper is concerned with the speaker diarization task in the specific context of the meeting room recordings. Firstly, different technical improvements of an E-HMM based system are proposed and evaluated in the framework of the NIST RT'06S evaluation campaign. Related experiments show an absolute gain of 6.4% overall speaker diarization error rate (DER) and 12.9% on the development and evaluation corpora respectively.

Secondly, this paper presents an original strategy to deal with the overlapping speech. Indeed, speech overlaps between speakers are largely involved in meetings due to the spontaneous nature of this kind of data and they are responsible for a decrease in performance of the speaker diarization system, if they are not dealt with. Experiments still conducted in the framework of the NIST RT'06S evaluation show the ability of the strategy in detecting overlapping speech (decrease of the missed speaker error rate), even if an overall gain in speaker diarization performance has not been achieved yet.

1 Introduction

For many years, the Rich Transcription task has evolved in terms of data processed - conversational telephone speech, broadcast news, meeting recordings - as well as the kind of information to be extracted - transcription, named entities, disfluencies, speaker information, ...

The speaker diarization task may be considered as taking part of this evolution. Also named "Who spoke When", this task consists in detecting the speaker turns inside an audio document (segmentation task) and grouping together all the segments belonging to a same speaker (clustering task). The difficulties of the task have also evolved with the different types of data processed. Initially applied to telephone conversational speech involving only 2 speakers in a constant acoustic environment, the difficulty of the task has been increased with the broadcast news, involving multiple speakers (up to 60 per hour), and multiple environments and speech quality (studio, telephone, interviews outside). Finally, the meeting room recordings, involving more spontaneous speech with large overlapping speech, speaker noise (laugh, whisper, coughing, ...), very short speaker turns, as well as a large variability in the signal quality due to the recording devices, have brought an additional level of complexity.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 359–370, 2006.

Since 2004, NIST has organized Rich Transcription evaluation campaigns focused on meeting room recordings. Numerous sites, mainly involved in a couple of projects (AMI [1], CHIL [2]), have been responsible for recording meetings, which are manually transcribed afterwards. The NIST RT'06S evaluation [3] is similar to the previous one on many points. Two sub-domain evaluations are still proposed, involving different speaker interactions: conference data, including real meeting recordings with 4 to 7 speakers, and lecture data including both lecturer and question/answer speech. Each sub-domain has different sensor types and setups (multiple distant microphones, microphone arrays, head microphones, etc), in relation with different tasks proposed to the participants. At novelty, the primary scoring metrics includes the overlapping speech this year (which was automatically discarded from scoring in the previous evaluations).

Derived from previous work [4][5][6], this paper presents some technical improvements brought to the LIA E-HMM based speaker diarization system. They were evaluated on the multiple distant microphone task of the NIST RT'06S evaluation. These technical improvements are concerned with both the E-HMM kernel algorithm as well as additional processes derived from the speaker recognition domain and proved to increase speaker diarization system performance [7]. Furthermore, the authors propose a first strategy to cope with the overlapping speech since it is one of the main focus of the NIST RT'06S evaluation as mentioned previously. This strategy aims at combining speaker information extracted from a virtual signal (built by mixing all the invididual microphone signals) and from the individual microphone signals themselves. The main idea is to use the virtual signal to extract a preliminary segmentation output and the individual channels, helped by the latter, to detect overlapping speech. This strategy was evaluated in the NIST RT'06S evaluation as well.

This paper is organized as follows: section 2 gives an overview of the baseline E-HMM based speaker diarization system. The technical improvements of the system are detailed in section 3, followed by the overlapping speech handling strategy in section 4. Experiments conducted on the different approaches are given in section 5, including the description of the experimental protocols, the results and related discussions. Finally, some conclusions are drawn in section 6.

2 Speaker Diarization System

The LIA speaker diarization system has been entirely developed by using the ALIZE toolkit [8], dedicated to speaker recognition (freely available thanks to an open software licence). The core of the system is built on a one-step segmentation algorithm implying an E-HMM (Evolutive HMM) [9]. Each E-HMM state characterizes a particular speaker and the transitions represent the speaker changes. All possible changes between speakers are authorized. In this context, the segmentation process is done in 4 steps:

1. **Initialization:** The HMM has only one state, called L_0, which the overall speech data of the audio file are attributed to. L_0 state is characterized by a GMM model, trained on all these speech data. Initially, it represents all

the speakers present in the audio file. In the next steps, these speakers will be moved one by one from L_0 state to new L_X states. At the end of the iterative process, speaker L_0 should represent only one speaker.

2. **New speaker added to the E-HMM:** A new speaker is selected and added to the E-HMM as follows: a segment is selected among all the segments belonging to L_0 according to the likelihood maximization criterion (see section 3.1 for more details). This selected segment is then used to estimate the GMM model of the new speaker named L_x added to the HMM.
3. **Adaptation/Decoding loop:** The objective is to detect all segments belonging to the new speaker L_x. All speaker models are re-estimated through an adaptation process according to the current segmentation. A Viterbi decoding pass is done, involving the entire HMM, in order to obtain a new segmentation. This adaptation/decoding loop is re-iterated while some significant changes are observed on the speaker segmentation between two successive iterations.
4. **Speaker model validation and stop criterion:** The current segmentation is analyzed in order to decide if the new added speaker L_x is relevant. In this case, the relevance is measured according to some heuristical rules on speaker L_x segment duration. If speaker L_x is considered as not being relevant, it is deleted and all its segments are given back to speaker L_0. The stop criterion is reached if there is no more 6 second long segment available in L_0 (to add a new speaker); otherwise, the process goes back to step 2.

Finally, a resegmentation process is applied, which aims at refining the boundaries and at deleting irrelevant speakers (e.g. speakers with too short speech segments). This stage is based only on the third step of the segmentation process. A HMM is generated from the segmentation and the adaptation/decoding loop is launched. At the end of each iteration, speakers with too short duration are deleted.

Concerning the front end-processing, the signal is characterized by 20 linear cepstral features (LFCC), computed every 10ms using a 20ms window. The cepstral features are augmented by energy. No frame removal or any coefficient normalization is applied at this stage.

The entire speaker segmentation process is largely described in [5].

2.1 Multiple Distant Microphones

This paper is only concerned with the multiple distant microphones located on the meeting room tables (MDM task of the RT'06S evaluation plan [3]). The number of individual channels is very variable from 2 to 7 or more according to the meeting room configuration. In this paper, the different channels are processed very simply by summing related signals in order to yield a unique virtual channel. No delay between signal nor enhancement of the channel quality is computed here.

2.2 Speech Activity Detection

Detecting speech and non speech segments is a crucial task for speaker diarization systems: misclassifying non speech segments as speech leads to an increase of

false alarm speaker error rate, involved in the scoring while misclassifying speech segments as non speech leads to an increase of missed speaker error rate.

In this paper, the speech activity detection (SAD) is handled by using a two state HMM, representing both speech and non speech events. 64 component GMM models are trained via an EM/ML algorithm for both states. Transition probabilities between states are fixed to 0.5. Finally, some duration rules are applied on segments in order to refine the speech/non speech segmentation yielded by the Viterbi decoding.

3 Technical Improvements

The analysis of the behavior of the E-HMM based system demonstrates that the one suffers from under-segmentation regarding the output segmentations. In some other words, the E-HMM system detects less speaker turn changes than expected in the references. This results in large segments, non homogeneous in terms of speakers which lead to an increase of the speaker diarization error rate. In this section, we propose three approaches to cope with this lack of segment homogeneity in terms of speakers.

3.1 Selection Technique

As described in section 2, the E-HMM scheme consists in adding a new speaker, involved in the conversation, at each new iteration until a stop criterion is reached. Adding a new speaker is the result of a selection procedure, which consists in searching, among the segments associated with L_0 speaker, a candidate segment representing the new speaker. Considering the baseline system, the selection technique was based on searching the best 6 second segment related to L_0 speaker according to a likelihood ratio maximization criterion as follows:

$$ArgMax_i[LL(S_i|M_{L_0}) - LL(S_i|M_{world})] \quad (1)$$

where $LL(S_i|M_{L_0})$ is the log likelihood of a 6 second segment S_i given the GMM model represented L_0 speaker, M_{world} a generic GMM model (also called world model) for log likelihood normalization.

As underlined by equation 1, this selection technique takes into account L_0 speaker only. However, assuming that new speakers have been already added to the HMM and that some data related to these speakers are misclassified and remain associated with L_0 speaker after the Viterbi decoding, the selection technique may select the best segment among these misclassified data. In this context, a new speaker will be falsely added to the HMM whereas it is already represented by another state. In some other words, a reference speaker will be split into two different hypothesis speakers in the final segmentation output, resulting in an increase of speaker segmentation error rate.

To cope with this main issue, we propose a selection technique involving all the speakers present in the HMM. The goal of this technique is to select a 6 second segment among the L_0 data, close to L_0 and far away from the other speakers according to the likelihood criterion. Here, equation 1 becomes:

$$ArgMax_i[(LL(S_i|M_{L_0}) - \frac{1}{N}\sum_{j=1}^{N} LL(S_i|M_{L_j})] \quad (2)$$

where M_{L_j} is the GMM model represented L_j speaker (different from L_0).

This discriminant selection technique is coupled with a frame selection inside the 6 second segments. Indeed, the split of L_0 data into 6 second segments is done in a sequential manner, without any control on the purity of segments. Therefore, the 6 second segments may contain L_0 speaker utterances and those of other speakers. By selecting frames inside each 6 second segment in the equation 2 permits to deal with this kind of non homogeneous segments. Indeed, the selection of frames is done separately when computing L_0 and L_j log likelihoods. Therefore, different sets of frames may be selected while dealing with L_0 and L_j models.

3.2 Segment Purification Scheme

As underlined in the previous section, one of the main issues of the E-HMM is to provide non-homogeneous segments in terms of speakers. In [10], a purification strategy based on a modified version of BIC criterion [11] is proposed to deal with non-homogeneous segments. Here, the authors propose an approach derived from this purification strategy, adapted to the E-HMM scheme. Applied before adding a new speaker in the E-HMM, this approach consists, for each speaker L_x (other than L_0), in:

- finding the best segment S_{Best} among the data associated with L_x speaker according to a likelihood ratio maximization criterion;
- comparing the best segment S_{Best} with all other segments S_x associated with L_x speaker according to the modified BIC criterion;
- if the BIC value between the best segment S_{Best} and a targeted segment S_i is above 0, segment S_i remains affected to speaker L_x otherwise segment S_i is moved to speaker L_0.

In this paper, the modified BIC criterion is applied by using GMM models of 5 components to characterize each source individually and 10 components to characterize both the sources together. Diagonal matrices are used for all the GMM models.

3.3 Post-normalization of Features

As demonstrated in the literature [7], the use of techniques issued from the speaker recognition domain may improve performance of speaker diarization systems. These techniques, such as the UBM-GMM based modeling [12], feature normalization (feature warping [13] or feature mapping [14]), are mainly used for the resegmentation phases in order to refine the output segmentation.

In this paper, the focus was made on the parameterization and feature normalization techniques. Some studies have shown that applying this kind of techniques earlier (e.g. during the speaker turn detection) does not help the segmentation process. Indeed, feature normalization for instance aims at reducing

the channel mismatch between training and testing data in speaker recognition systems. Conversely, in speaker segmentation, this kind of information may be helpful in the first steps for distinguishing speakers involving on different channels or environments. On the other hand, for the last steps of speaker diarization systems, like the resegmentation phase, feature normalization has been successfully used to improve the output segmentation accuracy.

Inspired from the LIA speaker verification system [15], the baseline speaker diarization system, described in section 2, was augmented with a second resegmentation phase, based on a different parameterization: 16MFCC, log energy, and their first derivatives. Moreover, the parameter vectors are normalized to fit a 0-mean and 1-variance distribution. The mean and variance estimators used for the normalization are computed segment by segment, using the output segmentation issued from the first resegmentation phase. The application of such a normalization technique at the segmental level permits here to estimate mean and variance on homogeneous data in terms of speakers (compared with an estimate on the overall audio file), since only one speaker tends to be present on each segment after the resegmentation.

4 First Proposal for Overlapping Speech Handling

As mentioned in the introduction, one of the main features of meeting data is the large amount of overlapping speech. Indeed, compared with broadcast news or conversational telephone data, meetings permit more natural and spontaneous speech. A couple of speakers may talk separately from the others in the room, speakers may interact more easily during discussions, they may laugh together as well. Therefore, the overlapping speech may represent a major part of a meeting. For the NIST RT'06S evaluation, it has been decided to take this overlapping speech into account in the scoring and to consider this as the primary task (compared with the previous evaluation plans, which simply discarded it).

In this paper, a first strategy to deal with the overlapping speech is proposed. The main idea is to combine both the information carried by the mixed channel signal (the unique virtual signal introduced in section 2.1) and the individual channel signals. First, the mixed channel signal is used to perform the diarization process, ignoring the overlapping speech. This first step leads to a non overlapping speech segmentation output. Secondly, the individual channel signals are processed in parallel, knowing this segmentation output. This aims at detecting segments on which two or more channels disagree in terms of speaker attribution.

From a technical point of view, the E-HMM based diarization system is first applied on the mixed channel signal in order to yield a first segmentation output. This first step aims at searching the different speakers talking during the meetings, assuming that they will be well detected in the non overlapping speech. Considering now that different speakers may talk together near to different microphones, processing the individual channel signals might permit to attribute an overlapping speech segment to different speakers. Therefore, the second step consists in applying the resegmentation step of the E-HMM based speaker diarization system on each individual channel signal, by using the segmentation

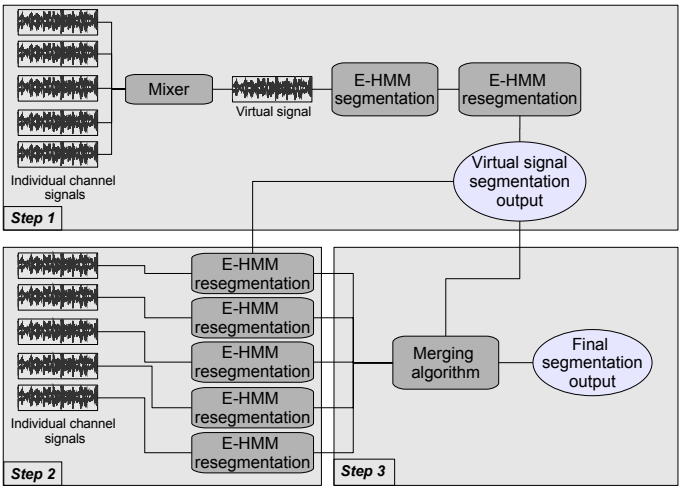


Fig. 1. Description of the overlapping speech strategy. Step1: E-HMM speaker diarization applied on the mixed channel signal (virtual signal). Step 2: E-HMM speaker resegmentation phase (initialized with mixed segmentation output) applied on each individual channel signal. Step 3: Merging of individual channel segmentation.

output issued from the virtual signal processing as initialization segmentation. Finally, the third step consists afterwards in merging all the individual channel dependent segmentation outputs plus the initial one issued from the virtual channel, in order to produce a final segmentation, which includes overlapping speech. This strategy is summarized in figure 1.

As mentioned previously, the goal of the merging algorithm is to produce a final segmentation by scanning all the segmentation outputs issued from the individual channel signal plus the mixed one. Two tasks are performed by this algorithm (illustrated in figure 2):

- to detect the disagreement zones between the different segmentation outputs, assuming that they are induced by overlapping speech. In this case, an additional segment is added in the final segmentation for each different speaker involved in the disagreement zones.
- to detect the agreement zones (segments with an identical speaker label over all the segmentation outputs), which have to be considered only once in the final segmentation to avoid redundant segments.

Limits of the Algorithm

The detection of zones of disagreement between the different channels is under the assumption that speakers talking simultaneously are closed to different microphones. Indeed, if a couple of speakers are talking together near the same microphone, the algorithm will not be able to separate both from only one source.

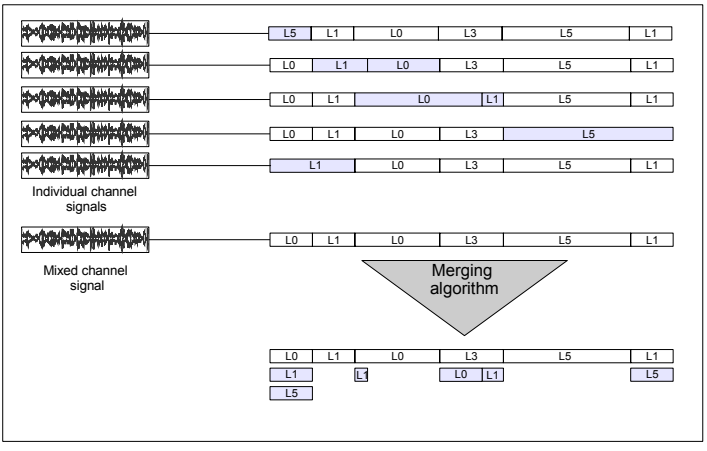


Fig. 2. Example of application of the merging algorithm, involved in the overlapping speech handling strategy

5 Experiments

5.1 Protocols

This work was done in the context of the RT'06S speaker diarization evaluation campaign [3]. Two sub-domain evaluations were still proposed as last year, involving, on the one hand, conference data issued from real meetings and lecture or seminar data on the other hand. These sub-domain evaluations are quite different in terms of the number of speakers involved in the audio files and of speaker turns. Here, both the speaker diarization and the speech activity detection systems were tuned on conference data only, because of their larger variability. Various corpora were used for the experiments:

- **Development corpus:** all the corpora made available for the previous evaluation campaigns are permitted for development purpose. Nevertheless, only data issued from the NIST RT'05S evaluation campaign (conference subdomain) were used here for the speaker diarization system development. These data include 10 meeting excerpts of about 12mn each, recorded on 5 different sites.
- **Training corpora:** for the speech activity detection system, we used sets of data issued from previous evaluation campaigns as well as data made available for training (conference and lecture/seminar sub-domain data) in order to train the speech and non speech models involved in the 2 state-HMM. For the speaker diarization system, we used an additional corpus made available for the Speaker Recognition Evaluation (NIST-SRE'05), including microphone recordings in order to train the world model required in the E-HMM adaptation process.

Table 1. Comparison of technical improvements brought to the baseline speaker diarization system, applied alone or combined, in terms of Diarization Error Rates (DER in %. Scoring limited to non overlapping speech) on conference data. (*Baseline*) baseline speaker diarization system, (*Selection*) selection technique, (*BIC*) segment purification scheme, (*Post-Proc*) post-normalization of features.

	DER in %	
System	Development set	Evaluation set
Baseline	22.5	39.4
Baseline+Post-Proc	18.1	32.0
Bas.+Selection	19.1	31.6
Bas.+Selection+Post-Proc	16.1	26.5
Bas.+BIC	20.8	37.2
Bas.+BIC+Post-Proc	17.9	27.7
Bas.+Selection+BIC	24.6	33.0
Bas.+Selection+BIC+Post-Proc	20.7	27.2

– **Evaluation corpora:** two corpora were distributed to the NIST RT'06S evaluation participants, one per sub-domain evaluation. For the conference sub-domain, 8 meeting excerpts of about 18mn each, recorded on 4 different sites were available. For the lecture sub-domain, 38 excerpts of 5mn each, recorded on 5 different sites were available.

In this paper, the performance of the speaker diarization system is expressed in terms of Diarization Error Rate (DER in %), including both the speech activity detection (through both missed speaker error and false alarm speaker error rates) and the speaker diarization system quality (through the speaker error rate) as detailed in [3].

5.2 Results and Discussion

This section presents the performance of the speaker diarization system. First, the different techniques presented in the previous sections are evaluated in terms of performance improvement, compared with the baseline system. This evaluation is reported on both the development and evaluation corpora. Secondly, experiments involving the speech overlapping handling strategy are presented.

Technical Improvements
Table 1 compares the different technical improvements brought to the baseline speaker diarization system: the selection technique (Selection), the segment purification scheme (BIC), and the post-normalization of features (Post-Proc). This comparison is done in terms of DER, on both the development and evaluation corpora. Here, the scoring does not take into account the overlapping speech. Different configurations are proposed in this comparison, combining the different techniques with the baseline system. Observing the results reached on the development data set, it can be pointed out that:

Table 2. Performance of the strategy proposed to handle the speech overlapping, expressed in terms of missed speaker error, false alarm speaker error, speaker error and overall diarization error rates (in %. Scoring including overlapping speech) on the evaluation data set

Approaches	Missed speaker error	False Alarm speaker error	Speaker error	DER
Baseline+Selection+Post-Proc	19.9	4.4	14.5	38.8
Baseline+Selection+Post-Proc+Overlap	17.6	8.9	14.5	41

- the combination of the post-normalization of features with any techniques leads to a decrease of DER (absolute gain from 2.9 to 4.4% depending on the techniques);
- the selection technique permits to improve the performance of the baseline speaker diarization system, with an absolute gain of 3.4% DER. A similar behavior is observed for the segment purification scheme, with a lower absolute gain of 1.7% DER;
- the combination of both the selection technique and the segment purification scheme degrades the performance of the baseline system;
- the best improvement is reached by combining the baseline system with both the selection technique and the post-normalization of features, with an absolute gain of 6.4% DER.

Regarding the evaluation data set, similar remarks may be drawn, with the best improvement reached by combining the baseline system with both the selection technique and the post-normalization of features as well, with an absolute gain of 12.9% DER.

These observations demonstrate that the selection technique and the segment purification scheme act at the same level. The analysis of the segmentation outputs shows that they permit to provide better outputs after the segmentation step, which may be more easily refined with the resegmentation phase. Conversely, the post-normalization of features acts in a different way, permitting an additional refinement of the segmentation outputs independently of the techniques previously applied.

Finally, the reproducible behavior of the different approaches over both the data sets demonstrates their robustness.

Overlapping Speech Handling Strategy

Table 2 compares the performance of the speaker diarization system with and without involving the speech overlapping handling strategy on the evaluation data set. As we can observe, the speech overlapping handling strategy permits to decrease the missed speaker error rate (absolute gain of 2.3%) while keeping the speaker error rate unchanged. Conversely, we can observe an increase of the false alarm speaker error rate (from 4.4 to 8.9%), leading therefore to an increase of the overall speaker diarization error rate (from 38.8 to 41%).

These observations tend to demonstrate that the strategy proposed succeeds in detecting overlapping speech (decrease of the missed speaker error rate).

Unfortunately, the strategy is strongly disturbed by the non speech segments misclassified as speech by the SAD system. Indeed, if a misclassified non speech segment is considered as an overlap zone and labelled as belonging to two different speakers during the individual channel processing, this segment will be counted twice as a false alarm speaker error. Therefore, the performance of the proposed strategy is partly correlated to the one of the SAD system.

6 Conclusion and Perspectives

This paper proposes three main technical improvements brought to the baseline E-HMM speaker diarization system, developed at the LIA in the framework of the NIST RT'06S evaluation campaign. The latter is focused on meeting room data collected on various sites. These improvements are concerned with the E-HMM kernel algorithm - selection technique and segment purification scheme - as well as the application of techniques issued from the speaker recognition domain - parameterization and segmental feature normalization. Experiments conducted on both development and evaluation data sets show that the parameterization and segmental feature normalization bring a significant decrease of speaker diarization error rate, compared with the baseline system. Moreover, the combination of the selection technique with the parameterization and segmental feature normalization reaches the best performance, with an absolute gain of 12.9% DER on the evaluation corpus, compared with the baseline system.

Secondly, the authors propose a novel strategy to deal with overlapping speech, which is largely present in meeting recordings and responsible for a decrease of performance of speaker diarization systems. The evaluation of this strategy on the NIST RT'06S data shows its ability to detect part of the overlapping speech (decrease of the missed speaker error rate, while the speaker error rate remains unchanged). Nevertheless, the behavior of the strategy on non speech segments misclassified as speech (due to the SAD system) leads to an increase of false alarm speaker error rates, and therefore an increase of the overall speaker diarization error rate.

Future work will consider other improvements of the E-HMM kernel algorithm, like the adaptation scheme of the GMM models. Regarding the overlapping speech based strategy, the issue involved by the SAD system has to be solved before pursuing and taking a real benefit in terms of performance increase.

References

[1] AMI: Augmented Multi-party Interaction project. (`http://www.amiproject org/`)

[2] CHIL: Computers in the Human Interaction Loop project. (`http://chil.server. de/servlet/is/101/`)

[3] NIST: Spring 2006 (RT'06S) Rich Transcription meeting recognition evaluation plan. `http://www.nist.gov/speech/tests/rt/rt2006/spring/docs/ rt06s-meeting-eval-plan-V2.pdf` (2006)

[4] Istrate, D., Fredouille, C., Meignier, S., Besacier, L., Bonastre, J.F.: NIST RT'05S evaluation: pre-processing techniques and speaker diarization on multiple microphone meetings. In: Machine Learning for Multimodal Interaction: Second International Workshop, Springer-Verlag, Edinburgh, UK. (2005)
[5] Meignier, S., Moraru, D., Fredouille, C., Bonastre, J.F., Besacier, L.: Step-by-step and integrated approaches in broadcast news speaker diarization. Special issue of Computer and Speech Language Journal, Vol. 20, Issue 2-3 (2006)
[6] Moraru, D., Meignier, S., Fredouille, C., Besacier, L., Bonastre, J.F.: The ELISA consortium approaches in broadcast news speaker segmentation during the NIST 2003 rich transcription evaluation. In: ICASSP'04, Montreal, Canada (2004)
[7] Zhu, X., Barras, C., Meignier, S., Gauvain, J.L.: Combining speaker identification and BIC for speaker diarization. In: EuroSpeech'05, Lisboa, Portugal (2005)
[8] Bonastre, J.F., Wils, F., Meignier, S.: ALIZE, a free toolkit for speaker recognition. In: ICASSP'05, Philadelphia, USA (2005)
[9] Meignier, S., Bonastre, J.F., Fredouille, C., Merlin, T.: Evolutive HMM for speaker tracking system. In: ICASSP'00, Istanbul, Turkey (2000)
[10] Anguerra, X., Wooters, C., Peskin, B., Aguilo, M.: Robust speaker segmentation for meetings: the icsi-sri spring 2005 diarization system. In: Rich Transcription 2005 Spring Meeting Recognition Evaluation, Edinburgh, Scotland (2005)
[11] Ajmera, J., Wooters, C.: A robust speaker clustering algorithm. In: ASRU'03, US Virgin Islands, USA. (2003)
[12] Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital Signal Processing (DSP), a review journal - Special issue on NIST 1999 speaker recognition workshop **10** (2000) 19–41
[13] Pelecanos, J., Sridharan, S.: Feature warping for robust speaker verification. In: 2001 : a Speaker Odyssey. The Speaker Recognition Workshop, Chania, Crete (2001) 213–218
[14] Reynolds, D.A.: Channel robust speaker verification via feature mapping. In: ICASSP'03 Conference, Hong Kong, China. (2003)
[15] Bonastre, J.F., Fredouille, C., Scheffer, N.: LIA 2005 system description. In: NIST SRE'05 Workshop: speaker recognition evaluation campaign, Montreal, Canada (2005)

The AMI Speaker Diarization System for NIST RT06s Meeting Data

David A. van Leeuwen[1] and Marijn Huijbregts[2]

[1] TNO Human Factors
Postbus 23, 3769 Soesterberg, The Netherlands
david.vanleeuwen@tno.nl
[2] Department of EEMCS, Human Media Interaction
University of Twente, Enschede, The Netherlands
marijn.huijbregts@utwente.nl

Abstract. We describe the systems submitted to the NIST RT06s evaluation for the Speech Activity Detection (SAD) and Speaker Diarization (SPKR) tasks. For speech activity detection, a new analysis methodology is presented that generalizes the Detection Erorr Tradeoff analysis commonly used in speaker detection tasks. The speaker diarization systems are based on the TNO and ICSI system submitted for RT05s. For the conference room evaluation Single Distant Microphone condition, the SAD results perform well at 4.23 % error rate, and the 'HMM-BIC' SPKR results perform competatively at an error rate of 37.2 % including overlapping speech.

1 Introduction

In the ongoing series of evaluations of speech technology the task of transcription and speaker diarization of meeting data has received renewed interest of several research groups, judged from the number of participants to the NIST Rich Transcription evaluation in Spring 2006 (RT06s) [1]. This paper describes the technologies employed in the Speaker Diarization (SPKR) and Speech Activity Detection (SAD) tasks by the AMI (Augmented Multiparty Interaction) team. This submission was the continuation of the RT05s work carried out by TNO [2], and was this year augmented by co-workers from the University of Twente. Further, in co-operation with AMI partners from the University of Edinburgh [3] an attempt was made to actually use the information of multiple microphones.

The task of Speaker Diarization in meetings can be summarized as a task to find out by automatic means 'who spoke when,' given the simultaneous recordings of several distant microphones located in the room. New challenges in the 2006 edition of RT included the scoring of overlapped regions of speech.

The AMI team tried to improve last year's approach by extending the BIC segmentation/clustering framework with Viterbi re-segmentation, and by exploring new clustering methods based on the work carried out by ICSI [4]. Further, some

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 371–384, 2006.

attempts were made to create systems that could produce overlapping speakers in the output.

During the workshop held to discuss the results of RT06s it turned out again how important it is to do speech activity detection properly before processing the microphone signal for further analysis (speaker diarization and speech-to-text). Although we employed a very basic SAD system, the performance was good for meeting room data, and we will therefore start the paper describing the SAD system, with training data bases and results. We will then continue by describing the three speaker diarization systems submitted to RT06s.

2 Speech Activity Detection

As pointed out earlier [2] the SAD task is implicitly important to the SPKR task due to the evaluation measure used in the SPKR task. In short, any period of the analysis that is mis-classified as either missed speech (miss) or, especially, false-alarm speech (FA) can only add to the speaker Diarization Error Rate (DER). Unless the SPKR system tries to deal with SAD itself, a badly performing SAD can be detrimental to the DER, even if the SPKR system itself functions very well.

The SAD system used here is a slightly improved version described for RT05s [2]. We use two Gaussian Mixture Models (GMMs) to model speech and silence, and use a Viterbi decoder to find the state sequence that maximizes the likelihood of the model sequence, given the data.

2.1 Databases

For the development of the SAD (and SPKR) systems, we exclusively used the RT05s evaluation meeting room data, henceforth named 'devtest' data. Only for the SAD system we used training material, for which we chose the 10 annotated AMI meetings distributed for RT05s development testing.

2.2 Microphone Conditions

Two sets of microphone conditions were investigated:

SDM. Single distant microphone. We used the microphone, as designated by NIST, without further pre-processing.

Array-beam. Multiple distant microphones with beam forming. For this condition we used the output of a delay-and-sum beam-former built by co-workers within the AMI project [3]. We used the output of the beam-formed acoustic signal as a single channel for further processing.

Note, that because our SDM initially gave us better devtest results than the beam-formed MDM data, we submitted both systems as (compulsory) 'MDM' condition, and designated our SDM as 'primary MDM.'

2.3 Features, Models, Decoder

This year we used RASTA-PLP features as extracted by ICSI's `rasta` tool [5] developed within the SPRACH project. Twelve PLP coefficients plus log energy were extracted over a window of 32 ms at a frame rated of 16 ms. First order derivatives of the 13 features were determined over 7 consecutive frames. The 26 features were modeled in a GMM with 16 components, trained with the 10 annotated AMI meetings. Two GMMs were thus trained, and a 2-state HMM network was built. Rather than training the transition probabilities $a(j|i)$, we took fixed values by setting:

$$\begin{gathered} \frac{a(\text{speech}|\text{sil})}{a(\text{sil}|\text{speech})} = R \\ a(\text{speech}|\text{sil}) + a(\text{sil}|\text{speech}) = P \\ \textstyle\sum_j a(j|i) = 1 \end{gathered} \tag{1}$$

and choosing $\log P = -30$ and $R = 10$. The parameters P and R were chosen to minimize SAD error for devtest data.

Last year's SAD system contained an error in the backtracking code of the Viterbi decoder, which had to be patched with a boxcar filter to smooth out state transitions. After removing this 'bug' from the system, post-filtering of the state sequence was no longer necessary. The only fenomenological filtering that we still applied was the removal of speech segments of duration shorter than 0.5 s, a filter that might not have been optimal if the SAD and SPKR error measures did not have defined a 'forgiveness' collar of 0.25 s around speech segments.

2.4 SAD Performance

In Table 1 we summarize the SAD results in terms of the Speech activity detection Error Rate on devtest and the RT06s evaluation data, the latter for both meeting room data and lecture room data. The SAD error rate is defined as

$$e_{\text{SAD}} = e_{\text{FA}} + e_{\text{miss}} = \frac{T_{\text{FA}} + T_{\text{miss}}}{T_{\text{speech}}} \tag{2}$$

where T_{FA}, T_{miss} and T_{speech} are aggregated times of silence misclassified as speech, speech misclassified as silence, and total duration of speech under evaluation.

A few observations can be made from Table 1. First, the single central microphone condition appears to outperform the array beam-formed signal processing slightly. This may seem due to the fact that the GMM models have been trained on SDM data only, but we have experimented with models trained on array beam-formed data as well, and could not obtain better results than the 3.43 % SAD error shown above. A second remark is that the performance for the Lecture room data is much worse than for the meeting room data, which will have its repercussions to the Speaker Diarization processing of the lecture room data that we present later. This is probably due to the out-of-domain model training,

Table 1. AMI SAD results for devtest and evaluation data, separated for SDM and MDM-array processing. Two sets of FA and miss values are given. The left set are time-weighted values where the *speech* evaluation time is in the denominator, the right set is determined using *non-speech* and *speech* evaluation time, respectively.

		(%)	prior (%)		no prior (%)	
Test set	Microphone	e_{SAD}	e_{FA}	e_{miss}	p_{FA}	p_{miss}
RT05s (devtest)	SDM	2.86	1.5	1.4	31.4	1.4
RT06s conference room	SDM	4.23	2.0	2.2	34.9	2.2
RT06s lecture room	SDM	26.0	6.3	19.7	45.7	19.7
RT05s (devtest)	Array-beam	3.41	2.6	0.8	54.1	0.8
RT06s conference room	Array-beam	4.82	3.8	1.0	66.3	1.0
RT06s lecture room	Array-beam	30.4	8.5	21.9	61.1	21.9

but it is rather worrying to observe that such an apparently simple task as SAD in our implementation is so sensitive to the acoustic domain it is applied to.

A more interesting observation we would want to make here, is the imbalance of FA and miss rates when expressed as detection tradeoff measures $p_{\mathrm{FA}} = T_{\mathrm{FA}}/T_{\mathrm{sil}}$ and $p_{\mathrm{miss}} = T_{\mathrm{miss}}/T_{\mathrm{speech}}$, where T_{sil} is the actual duration of silence in the evaluation. At the workshop following the RT06s evaluation, it was noted that within the CHIL project, the evaluation measures SDER/NDER are defined as p_{FA} and p_{miss} [6]. These measures are insensitive of the evaluation prior $p(\mathrm{speech})$, and in a sense provide better indication of the discriminability of the detector than e_{SAD}. The latter measure is, for a given detector, dependent on the prior $p(\mathrm{speech})$. It is questionable to optimize a detector using an evaluation measure that includes the prior by minimizing e_{SAD}, as we did by choosing a particular value for R.

In speaker detection, which is also in essence a two-class discrimination problem, it is customary to plot the tradeoff between p_{FA} and p_{miss} in a Detection Error Tradeoff plot [7], a ROC-curve on axes scaled by the probit function [8]. The basis of a DET-analysis is formed by two sets of trials (*target* and *non-target*) with *scores* indicating the support for a trial being a target-trial. The fact that a score is used means that the detector does not make actual *decision* for this analysis, and the decisions can be postponed to the moment of plotting (every point on a DET curve corresponds to a given threshold). We would like to generalize this concept to a time-based segmentation where, by segmentation, decisions *have* been made.

2.5 Time-Weighted DET Curve

We require our detector to not only produce a segmentation in time, but to also give a *likelihood score* for each segment that that segment is speech (the target class). In comparing the hypothesis segmentation with the reference segmentation we can obtain a higher-grained 'evaluation segmentation' of segments that are either correct speech, correct silence, FA, or miss segments. These segments can now be seen as trials, either belonging to target (speech) or non-target (silence) class, depending on the reference segmentation, each with a score. Let

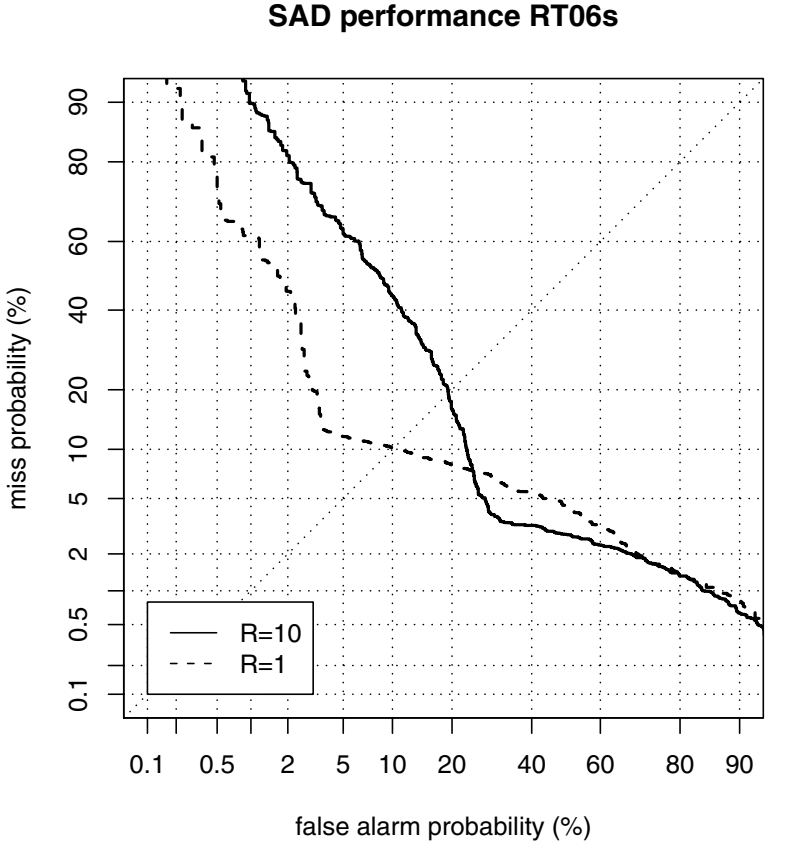


Fig. 1. Time-weighted DET plot for the RT06s meeting room data. The solid line has a transition odds $R = 10$ in Eq. 1, the dashed line $R = 1$. Note that the DET axes have been extended to 90 % error probability.

T_i be the durations of these evaluation segments, and the scores be s_i, then for obtaining a point on the DET curve we can set a threshold θ, from which we define the FA probability

$$p_{\mathrm{FA}}(\theta) = \sum_{i \in \mathcal{S}_{\mathrm{non}}, s_i > \theta} T_i \Big/ \sum_{i \in \mathcal{S}_{\mathrm{non}}} T_i, \tag{3}$$

and correspondingly the miss probability

$$p_{\mathrm{miss}}(\theta) = \sum_{i \in \mathcal{S}_{\mathrm{target}}, s_i < \theta} T_i \Big/ \sum_{i \in \mathcal{S}_{\mathrm{target}}} T_i. \tag{4}$$

Here $\mathcal{S}_{\mathrm{non,target}}$ denote the set of evaluation segments that are actually non-target and target class, respectively, according to the reference. By varying θ,

the time weighted DET-plot $\left(\text{probit}\, p_{\text{FA}}(\theta), \text{probit}\, p_{\text{miss}}(\theta)\right)$ can be constructed. As for traditional DET plot, this process can be calculated efficiently by sorting segments by score and computing FA and miss time cumulatively.

In order to investigate the feasibility of this new analysis, we generated scores by averaging the log likelihood ratios of the speech and silence models over the segments found by the decoder. In Fig. 1 the corresponding time-weighted DET plot is shown for the RT06s meeting room data. The two curves show different settings for the parameter R from Eq. 2. The solid line shows the curve for the submitted parameter setting $R = 10$ that minimized e_{SAD} for devtest data, the dashed line is an contrastive setting $R = 1$. Both curves show a little 'knee bend' around the operating point that corresponds to the *actual* decisions that were made in the segmentation process. The dashed curve shows better general detection capabilities, but happens to perform slightly worse around the optimum operating point defined by the prior p(speech).

3 Speaker Diarization Systems

For the speaker diarization task we submitted three bottom-up segmentation/ clustering systems. The first was based on a single consecutive BIC segmentation and clustering, the remaining two on iterative Viterbi segmentation and clustering. All system start with the output from the SAD systems described above.

3.1 BIC Based Segmentation and Clustering

The first speaker diarization system is a continuation of the TNO system in RT05s [2]. The Bayesian Information Criterion (BIC) is used to first segment the speech in speaker-homogeneous segments, which are then later clustered based on a Gish distance measure and using a BIC stop criterion. An addition this year was a final Viterbi re-segmentation based on 16-component GMM models that are trained on the clusters found in the BIC clustering process. In this re-segmentation process the silence regions found by the SAD are included to train a separate silence GMM model, and this model is included in the Viterbi segmentation.

One of the less satisfactory properties of the BIC method is that it has two tunable parameters λ_s and λ_c for the segmentation process and the stop-criterion, respectively. Merely by manual tuning we could improve devtest DER, not scoring overlapping speech, $e_{\text{SD}}^{\text{no}}$ from 34.2 % using $\lambda_s = 1.5$, $\lambda_c = 14$ (the parameters used for the RT05s evaluation) to 25.4 % by using $\lambda_s = 1.8$, $\lambda_c = 6$. The additional Viterbi re-segmentation further lowered $e_{\text{SD}}^{\text{no}}$ to 21.7 %.

The fact that the devtest results were so dependent on the two λ parameters did not give us very much confidence that the parameter setting would hold for RT06s, and as can be seen from Table 2 the evaluation results were rather poor, which is why we moved to better approached discussed below in section 3.2.

From the table an enormous degradation of performance can be observed going from devtest to the evaluation, specifically for the beam-formed microphone data.

Table 2. Results of the BIC-based speaker diarization system for development test and evaluation test, both for evaluation condition without and with overlapping speech regions. For later reference, the processing speed (measured on an AMD Opteron 250 CPU) is indicated for the evaluation data.

Test set	Microphone	$e_{\mathrm{SD}}^{\mathrm{no}}$ (%)	e_{SD}^{o} (%)	Speed (× RT)
RT05s conference room	SDM	21.7	29.4	
RT06s conference room	SDM	32.4	44.8	0.79
RT06 lecture room	SDM	26.2	27.3	0.14
RT05s conference room	Array-beam	19.7	27.5	
RT06s conference room	Array-beam	35.6	46.3	
RT06 lecture room	Array-beam	47.6	48.7	

Dealing with Overlap. New in RT06s was that the primary evaluation measure included speech regions with overlapping speakers. This means that an ideal system should be capable of producing output where speakers overlap. Clearly, in the plain segmentation and clustering approach, there is at most one speaker in the output. We have experimented with a few algorithms to deal with this challenge. None of them resulted in lower e_{SD}^{o} on the devtest, but we want to briefly describe the idea.

Average overlap. A trivial approach, with the sole purpose of potentially lowering the error rate, is based on the expectation that in the turn-taking process speakers may, on average, overlap. A trivial implementation is to post-process the output of a one-speaker diarization system such that boundaries between speakers are altered so that for a short time t_o speakers overlap. This t_o would represent the average time of overlap between speakers, and may be tuned to give minimum e_{SD}^{o}. We found $t_o = 0$.

Two-speaker states. A more interesting approach assumes that the features of the sum of two speech signals can be approximated by the sum of the features of the individual signals. Then starting with a segmentation of a one-speaker diarization system, we can build N state models for the individual clusters, and then add $\binom{N}{2}$ '2-speaker states,' which are trained using speech from pairs of clusters. The HMM topologycan then be extended with transitions from single speaker states to 2-speaker states that include that speaker, and vice versa. In decoding with this more complex model, the hope is that the GMMs of the 2-speaker state get activated when its speakers speak simultaneously. Although our implementation did produce overlapping speech, we observed that only in approximately 1/3 of the overlap time these were the correct speakers, while in the remaining cases the wrong speakers were produced. Hence, we could not lower e_{SD}^{o} this way.

3.2 Decoder Based Speaker Diarization

In addition to the full-covariance single Gaussian BIC-based system, two decoder-based speaker diarization systems were submitted. Both contrastive

speaker diarization systems consist of two modules. First the SAD module described in Sect. 2 filters out the non-speech parts of the audio signal. The second module, a Viterbi decoder, segments and clusters the speech, using diagonal covariance GMMs.

The decoder is an adapted version of the University of Twente 2006 decoder (UT06). The UT06 decoder is a Large Vocabulary Continuous Speech Recognition (LVCSR) decoder using Hidden Markov Models (HMM). Its state emission probabilities are calculated by Gaussian Mixture Models (GMM). The decoder uses the Perceptual Minimum Variance Distortionless Response (PMVDR) cepstral coefficients from the Sonic LVCSR toolkit [9]. For speaker clustering, the derivatives and energy features are not used and the decoder's lexical tree is replaced by a network of single state HMMs connected to each other by a single (non-emitting) start- and end-state. In the ideal situation, each state is trained on audio of exactly one unique speaker so that a final Viterbi alignment would result in an optimal segmentation and clustering of the speech.

In order to find an approximation to this ideal HMM topology, a straightforward algorithm is used. Initially the number of parallel states is chosen higher than the expected number of speakers. For conference meetings ten states are used. For lecture meetings the initial number of parallel states is five. The audio is randomly divided over the parallel states so that the HMM can be trained for the first time. In an iterative process, each state will split its Gaussian with the highest weight until the desired number of initial Gaussians is reached. This initial number of Gaussians is dependent on the total amount of data that is used to train the state. Experiments on the devtest data showed that the optimum number of training samples per Gaussian is 800. For the RT06s conference meetings, this means that each state is initially trained with approximately 10–14 Gaussians. Making the number of Gaussians dependent on the duration of the meeting (the number of training samples) will ensure that the states are not under- or over-trained when the duration of the audio varies.

After the initial training iteration, a Viterbi alignment re-segments the audio for a new training iteration. This procedure is repeated until the overall Viterbi likelihood score does not improve any more. During this procedure, the states that initially were trained with data from various speakers will gradually change into states that are trained with data from a single dominating speaker. Because of small initial differences in the speaker distribution of each state, at the first iteration each state will be trained slightly better for one of the speakers compared to the others. By re-aligning the data, the states will attract data from these speakers. Each iteration the states will be trained on more data from the dominated speaker and on less from the other speakers.

Once each state is trained for the most part on a single speaker, the task remaining in order to obtain the ideal HMM topology is to reduce the number of states until there is exactly one state left for each speaker. The two systems use different methods to reduce the number of states. The following paragraphs will describe these methods.

HMM-BIC. The HMM-BIC training approach is inspired by the ICSI-SRI Spring 2005 Diarization System described in [4]. The number of states is reduced by pairwise merging two states into one single state. The two states to merge are found by using the Bayesian Information Criterion (BIC).

For each combination of two states, a new state σ is trained containing the sum of the number of Gaussians in the two original states σ_a and σ_b. This *merged* state is trained using the training data of the original two states. From the Viterbi scores of these three models, the BIC score is calculated. Because the total number of Gaussians will not change if σ replaces σ_a and σ_b, the complexity of the system will not change. This makes it possible to calculate the BIC score for merging two models without the need for a penalty for model complexity which obsoletes the necessity for the parameter λ [10]. Let D_a be the data used to train model σ_a, and let D_b to train σ_b and D to train model σ, then

$$\mathrm{BIC}(\sigma_a, \sigma_b) = \log P(D|\sigma) - \log P(D_a|\sigma_a) - \log P(D_b|\sigma_b). \tag{5}$$

The BIC value is calculated for all possible pairs of states. If none of the BIC scores are higher than zero, merging is stopped. Otherwise the two states with the highest BIC score are merged. This effectively means that the two states are removed from the network and the merged state is placed into the network. The data of each speech segment (from the SAD) is re-aligned using this new network and another training iteration is performed. After that, the merging procedure is repeated.

Cut&Mix. Considering the merging of all combinations of two states, as described in the previous paragraph, takes a lot of computational effort. Another disadvantage of the HMM-BIC method is that it is based on the assumption that the two states are trained with data from the same single speaker. Unfortunately when a state contains data from multiple speakers, this data is not spread over multiple states when the state is taken out of the system. The method of reducing states described in this section is developed in order to make the system faster and to make it more robust in cases a state is not trained solely using one speaker.

The basic idea of the Cut&Mix strategy is that all states are considered for removal, and that the state which improves the Viterbi likelihood most after removal is subsequently 'cut out,' and its Gaussians are redistributed over the remaining states.

Formally, at each iteration i the number of states in the HMM will be reduced by one, by sequentially removing state j. The overall Viterbi score L will be used to compare the original HMM topology $\mathcal{T}_i$ with the new topology $\mathcal{T}_{i+1}^j$. If the score $\max_j L(\mathcal{T}_{i+1}^j)$ is higher than the original score $L(\mathcal{T}_i)$, the new topology $\mathcal{T}_{i+1}^j$ is used in the next iteration as $\mathcal{T}_{i+1}$. If the maximum is lower than $L(\mathcal{T}_i)$, the new topology is discarded and the original HMM will be regarded as the optimum topology.

Once state j is to be removed from the topology, the number of Gaussians in the remaining states is increased until the total number of Gaussians in the original topology and in the new topology are equal.

In order to make a fair comparison between the Viterbi score of the original system $\mathcal{T}_i$ and the new HMM topology $\mathcal{T}_{i+1}$, the complexity of both systems should be the same. Therefore, the number of Gaussians of the new system should be increased until it is the same as the number of Gaussians in the original system. The Gaussians from the state that has been cut away are distributed over the GMMs in the same proportions as the speech data has been distributed by the Viterbi alignment. For each state in the HMM, the increase in assigned data from before cutting away state j and after cutting away state j is calculated. Also, the amount of data that was originally assigned to state j is divided by the number of Gaussians in this state. The result is the amount of increased data that a state needs in order to be assigned an extra Gaussian. If $N(x, i)$ is the amount of data (speech frames) used to train state x at $\mathcal{T}_i$ and $n(x)$ is the number of Gaussians in state x, then the number of extra Gaussians $\Delta n(k)$ for each state k at $\mathcal{T}_{i+1}$ is:

$$\Delta n(k) = \frac{N(k, i+1) - N(k, i)}{N(j, i)/n(j)} \tag{6}$$

Each GMM that is assigned new Gaussians will be re-trained. The new Gaussians are obtained by splitting the Gaussian with the highest weight. After all GMMs are re-trained, the new overall Viterbi score is calculated. This score will be compared to the original score. If this score is higher than the original, a new cutting iteration will be started.

RT-06 Results. The speaker diarization error rates of the two decoder-based systems on the conference meeting audio are listed in Table 3. These systems only use the SDM microphones. On the devtest set, the data from RT05s, the Cut&Mix system outperforms the HMM-BIC system. This is not reflected in the results of the evaluation. As expected, the processing speed of the Cut&Mix system (real-time factor 2.25) is better than the speed of HMM-BIC (4.63). These factors are measured on an Intel Xeon 2.8 GHz processor.

Post Evaluation Analysis. During state reduction, the HMM-BIC method calculates all possible topologies with one less state. Once two states are merged, no further training of the HMM is needed and, using BIC, the best topology is chosen. The Cut&Mix method does not calculate all topologies before choosing the state to remove from the HMM. Although this method uses a good measure

Table 3. The speaker diarization results of the HMM-BIC and Cut&Mix decoder based systems

		HMM-BIC		Cut&Mix	
Test set	Microphone	$e^{\text{no}}_{\text{SD}}$ (%)	e^{o}_{SD} (%)	$e^{\text{no}}_{\text{SD}}$ (%)	e^{o}_{SD} (%)
RT05s conference room	SDM	21.6	30.2	18.6	27.6
RT06s conference room	SDM	22.7	37.2	25.2	39.5
RT06 lecture room	SDM	30.8	32.4	30.1	31.6
Processing speed (×RT)		4.63		2.25	

to pick the best state to remove, the remaining states need training after the choice has been made and only afterwards it can be determined if the new topology is indeed better than the original topology. When the original topology turns out to be better, the system will stop without considering cutting other states out of the HMM. Therefore it is possible that the system stops reducing states too soon and that the clustering result contains too many speakers.

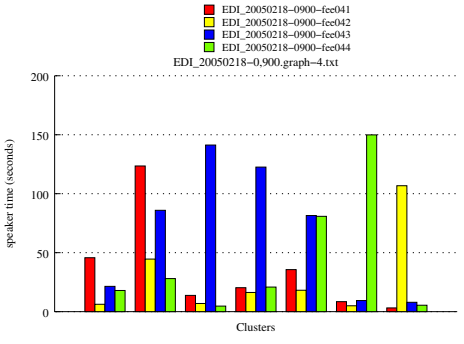


Fig. 2. Final speaker distribution of meeting EDI-20050218

Figure 2 is a graphical representation of a situation where this is the case. It shows seven clusters (states) that contain the amount of speaker time of each speaker in the meeting that is assigned to the cluster. During the final attempt to reduce states, the first cluster was chosen to be cut away. This resulted in a worse performing topology and therefore the seven clusters were maintained. Cutting away the third or fourth state probably would have resulted in a better topology. The speaker diarization error-rate without overlapping speech of the HMM-BIC system on this meeting was 29.5%. On the Cut&Mix system this was 44.9%. We believe that this shortcoming in the stop criterion of the Cut&Mix system is the main reason why it did not perform as expected.

For both systems, after the initial HMM training iteration, each state should contain a dominating speaker. If this is not the case, reducing the states will not always result in a clean final clustering. States trained on multiple speakers may remain in the system (like for example the fifth cluster in figure 2) and speakers that were not dominant in any state during the initial training iteration, may never earn their own state. In Fig. 3 the speaker distribution in one of the conference meetings after the initial training iteration is drawn (graph at the left). Speaker 'vhqqmy' (the rightmost bar in each cluster) is not dominating in any of the clusters. The graph at the right in figure 3 shows the speaker distribution after the final iteration. At this point speaker 'vhqqmy' is still not assigned its own state. Also, the first two states contain data from multiple speakers. It is possible that states with multiple speakers contain a considerable amount of overlapping speech. This would explain why data from these states is not assigned to states with dominating speakers (e.g., the fourth state in Fig. 3,

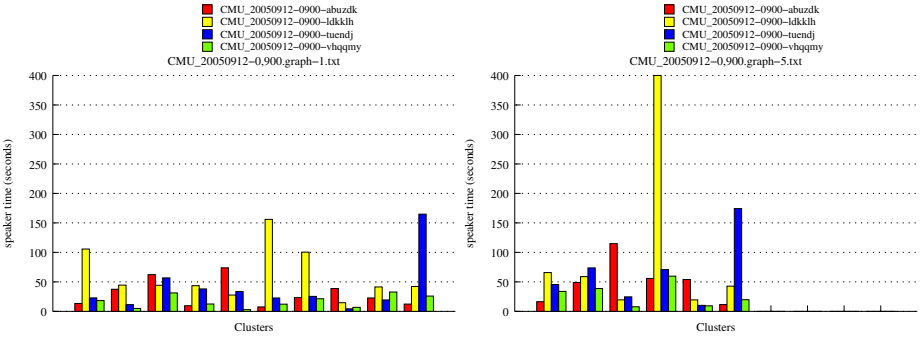


Fig. 3. Speaker distribution of meeting CMU-20050912. Left: distribution after the first training iteration. Right: distribution after the final training iteration.

right). Apart from this problem, a better method for determining the initial speaker distribution might improve system performance considerably.

4 Discussion and Conclusions

The performance of our SAD system, when expressed in time-weighted error terms e_{SAD}, seems consistently performing very well for the meeting room evaluation data. For both RT05s and RT06s this relatively straightforward system showed low error rates. As a pre-processing step to a SPKR system it functions adequately. Our AMI colleagues [3] have tried to use the SAD output for the speech-to-text MDM task, but this lead to practical problems such as a wide range of segment sizes that were difficult to deal with. Note that all evaluation measures discussed here for SAD and SPKR tasks do not evaluate the frequency at which (erroneous) segment boundaries occur. Very many spurious error segments, if small in duration, will not add much to the error rates, while such a result may be completely useless for interpretation by person or machine.

The evaluation prior p(speech) appears to be very important in choosing the operating point for the SAD decoder. In the devtest set this prior was 0.955, for RT06s it turned out to be 0.947, very close. In speech recognition, another field where decoding plays an essential role, it is normal to include prior information in the system (e.g., the language model), while for speaker detection it is customary to take the prior out of the detector. When SAD is seen as a detection problem, it might be more diagnostic to work with detection-based evaluation measures, such as p_{FA}, p_{miss} and analyze the detector in terms of post-evaluation measures such as the Equal Error Rate. We have shown that the DET trial-based framework can be extended to a time-weighted segmentation-based framework. Our first experiments show that the obtained DET curves look reasonable (a 'straighter' DET curve indicates that the underlying score distributions are 'more Gaussian' [11]), but changing detection thresholds after scores have been produced is not the same as choosing different segmentation parameters, such as our R. In a way, this approach resembles the evaluation

of word spotting systems, where also segmentation (the spotting of words) is carried out including the production of scores (acoustic likelihoods).

The SAD performance for lecture room meetings, however, was far below average. We have suggested that this may be the result of acoustic mismatch of the models. This may have been the cause of our poor SPKR results for the lecture room meetings, but we are not convinced that the lecture room domain is an interesting problem for speaker diarization, see for instance the 'one speaker takes all' approach of ICSI for RT05s [4].

In our development of different SPKR systems, it has become clear that the popular BIC segmentation method, originally developed for Broadcast News domains, shows severe problems in terms of the tuning of the complexity penalty parameter λ. The two approaches based on the ICSI system of RT05s, the standard HMM-BIC and the derived Cut&Mix systems use Gaussian mixtures to keep the complexity of the system constant so that the systems do not need tunable parameters. These systems have proven to be more robust against new evaluation data. One might argue that the single Gaussian BIC segmentation method still has advantages, such as computational efficiency. The system runs easily under $1 \times$ RT, the figures reported in Table 2 are high estimates because we ran our system for the entire meeting, rather than only the segments indicated in the evaluation index files. However, we believe that the sensitivity of the choice of the λ to the application domain needs proper attention.

Concentrating now on the two decoder-based speaker diarization system, the RT06s evaluation has shown that our HMM based approach is competitive to other systems. The Cut&Mix system is not performing as well as the HMM-BIC system, probably because of its poor stop criterion. We are planning to test other stop criteria for the Cut&Mix system in order to improve its performance without increasing the processor load.

Analysis of the SPKR evaluation results show that the initial speaker distribution affects the final distribution. Speakers that are not dominating any state in the initial training run will not likely be assigned a private state during the following iterations. Some states that contain about the same amount of data of multiple speakers will not converge to a single speaker. In future we will investigate if this problem is caused by overlapping speech. Also we will investigate new methods for distributing data for the initial training iteration.

Acknowledgements

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication). It is further partly supported by the project MultimediaN (`http://www.multimedian.nl`). MultimediaN is sponsored by the Dutch government under contract BSIK 03031.

References

1. Fiscus, J.G., Radde, N., Garofolo, J.S., Le, A., Ajot, J., Laprun, C.: The rich transcription 2006 spring meeting recognition evaluation. Lecture Notes in Computer Science (2007) 309–322

2. van Leeuwen, D.A.: The TNO speaker diarization system for NIST rich transcription evaluation 2005 for meeting data. Lecture Notes in Computer Science (2006) 400–449
3. Hain, T., Burget, L., Dines, J., Garau, G., Karafiat, M., Lincoln, M., Vepa, J., Wan, V.: The AMI meeting transcription system: Progress and performance. Lecture Notes in Computer Science (2007) 419–431
4. Anguera, X., Wooters, C., Peskin, B., Aguiló, M.: Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system. Lecture Notes in Computer Science (2006) 402–414
5. Hermansky, H., Morgan, N.: Rasta processing of speech. IEEE Transactions on Speech and Audio Processing, special issue on Robust Speech Recognition **2** (1994) 578–589
6. Macho, D., Temko, A., Nadeu, C.: Robust speech activity detection in interactive smart-room environment. Lecture Notes in Computer Science (2007) 236–247
7. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The DET curve in assessment of detection task performance. In: Proc. Eurospeech 1997, Rhodes, Greece (1997) 1895–1898
8. van Leeuwen, D.A., Bouten, J.S.: Results of the 2003 NFI-TNO forensic speaker recognition evaluation. In: Proc. Odyssey 2004 Speaker and Language recognition workshop, ISCA (2004) 75–82
9. Pellom, B., Hacioglu, K.: Recent Improvements in the CU Sonic ASR system for Noisy Speech: The SPINE Task. In: Proc. ICASSP. (2003)
10. Ajmera, J., McCowan, I., Bourlard, H.: Robust speaker change detection. IEEE Signal Processing Lettres **11** (2004) 649–651
11. Navrátil, J., Ramsawamy, G.N.: The awe and mistery of t-norm. In: Proc. Eurospeech. (2003) 2009–2012

The 2006 Athens Information Technology Speech Activity Detection and Speaker Diarization Systems

Elias Rentzeperis, Andreas Stergiou, Christos Boukis,
Aristodemos Pnevmatikakis, and Lazaros C. Polymenakos

Autonomic & Grid Computing Group
Athens Information Technology
Athens, Greece

Abstract. This paper describes the Speech Activity Detection (SAD) and Speaker Diarization (SPKR) systems that were developed by the Athens Information Technology in the scope of the NIST RT-06S evaluations. The SAD system performs classification of recorded frames into speech and non-speech, using Linear Discriminant Analysis (LDA), while the SPKR one initially segments recordings into speech intervals based on the Bayesian Information Criterion (BIC), and then applies a two-step clustering strategy to group segments from the same speaker together. Following a discussion of the intrinsics of the two systems, we report and comment on our results on the RT-06S corpus [20].

1 Introduction

Several years after the introduction of the ubiquitous computing vision [1], we are witnessing the emergence of a host of computing applications that provide context-awareness through implicitly deriving information about the surrounding environment. Key to deriving this context is the availability of perceptual components processing audio-visual streams. While context comprises a rich set of information elements, knowledge relating to location, time and activity appears to be more important, since it may act as an anchor to extract additional cues. Context aware appliactions can significantly enhance human-computer interaction, giving rise to a computing paradigm shift that places humans, rather than machines, at the heart of the human-computer interaction. In this paradigm computers provide continuous support to humans through recognizing people and their goals, anticipating their needs and accordingly facilitating their tasks.

The EU-funded project CHIL (Computers in the Human Interaction Loop) [2] follows along these lines with the goal of '... fading the computers in the background ...' in order to realize the concept of ambient intelligence and provide perceptual technologies as well as services that can support smart spaces in both home and office environments. The members of the CHIL consortium have setup the so called 'smart labs' equipped with a multitude of audio-visual

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 385–395, 2006.

sensors (cameras and microphones) towards answering the three fundamental questions of 'Where' (determine the location of an event of interest taking place in a room), 'Who' (determine the identities of the people participating in this event) and 'What' (identify the current activity undertaken by the people in the room) in an unobtrusive manner. The answers to the first two of these questions can be obtained by low-level perceptual component processing, whereas the detection and classification of activities must be carried out at a higher level and usually relies on rule-based learning, as well as the notion and definition of states and transitions between them [3].

At Athens Information Technology, we have developed a series of components that answer the questions of 'Where' and 'Who' using audio cues only, as have been demonstrated in the recent CLEAR 2006 evaluation campaign and workshop [4]. The question of 'Where' is answered by some form of tracking or surveillance, which can be performed in 3D with an appropriate microphone setup [5], whereas that of 'Who' is dealt with by means of a speaker identification system [6], which can again take advantage of multiple microphones to improve recognition accuracy, as illustrated in [7]. Despite the fact that the actual recordings were by no means constrained, the CLEAR evaluation plan imposed some caveats on the speaker identification task, namely that each testing segment contained only speech (i.e., no non-speech sounds or silence) which was uttered by a single person out of a predefined database of known speakers. This gives rise to the following issues:

- What happens when there are non-voiced sounds (coughs, keyboard clicks etc.) or silence mixed with the actual utterances?
- How can we process a recording in which some unknown speakers (the number of which is not known a priori) interact freely (i.e., interrupt each other and may even talk concurrently for some time interval)?

To answer these questions, we have implemented both a Speech Activity Detection system, which can discard non-voiced sounds and silence intervals and forward only speech segments to either a tracker or a recognizer, as well as a Speaker Diarization module that breaks up a recording into turns and essentially answers the 'Who Spoke When' question. These two systems, as well as their performance in the NIST RT-06S evaluation, are the focus of this paper, which is further organized as follows: sections 2 and 3 describe the details of our current Speech Activity Detection and Speaker Diarization implementations, whereas section 4 discusses how they fared in the respective tasks of the RT-06S evaluation under different types of recordings and sets of employed microphones used for input. Finally, section 5 concludes the paper and discusses future extensions to these systems.

2 Speech Activity Detection

The performance of several acoustic perceptual components like speech recognition, person localization or speech coding is enhanced to a great extent if there is

a mechanism that can successfully separate the speech from the non-speech segments. This holds especially in noisy or reverberant environments. The need for this preprocessing step has led to the development of several SAD systems. The use of such systems has led not only to the improvement of the success rates of audio processing applications, but also to significant computational power reduction or improvement of channel capacity in the case of cellular telephony.

Several SAD algorithms have been proposed in the literature. These attempt to separate speech from non-speech based on different metrics like signal energy and zero-crossing rate measurements [8], statistical modeling [9], wavelet transform [10] and adaptive thresholds [11]. In this paper, embarking upon [12] we propose an algorithm that applies LDA [13] to speech data that have been transformed to Mel Frequency Cepstral Coefficients (MFCC, [14]). The decision is based on simple thresholding.

2.1 Overview of the Components

The proposed algorithm consists of two fundamental components, namely the MFCC and the LDA method.

MFCC are the dominant features used for speech recognition. The filters utilized for the calculation of the MFCCs are motivated by the response of the human auditory system and thus spaced linearly at low frequencies and logarithmically at high ones. As a result the MFCCs manage to accurately capture the phonetically important characteristics of a speech signal.

LDA is a linear subspace projection method that maximizes the between-class scatter of the data under the constraint that the within-class scatter is minimum. Hence, clusters that correspond to classes are formed, and they are as far apart from other classes as possible, yielding a subspace suitable for classification. In the case of SAD there are two classes to be discriminated, speech and non-speech. Since LDA reduces the dimension of the projected data to the number of classes minus one, the projected values of speech and non-speech are one-dimensional. Figure 1 shows the histograms of the projected values of speech and non-speech data for a sample recording. Even though there is some overlap between the two classes, the vast majority of the data are well separated.

2.2 System Overview

The system is allowed to have a priori knowledge of the data collection site, room configuration and sensor types. To that effect, separate training for every site and room was adopted. For instance the training parameters of the testing data of site Y were evaluated from the development data of site Y or from a site that had similar room configuration and sensor types to Y, whenever development data from Y were not available or were not sufficient in quantity. We found that this approach yielded considerably better results that a globally trained system, mainly due to the differences in sensor types, numbers and layouts between conference meeting and lecture meeting rooms.

Audio inputs were collected from at least 3 omni-directional microphones placed on a table in between the speakers. In order to suppress noise and emphasize the speech intervals, spatial averaging of the audio inputs was performed.

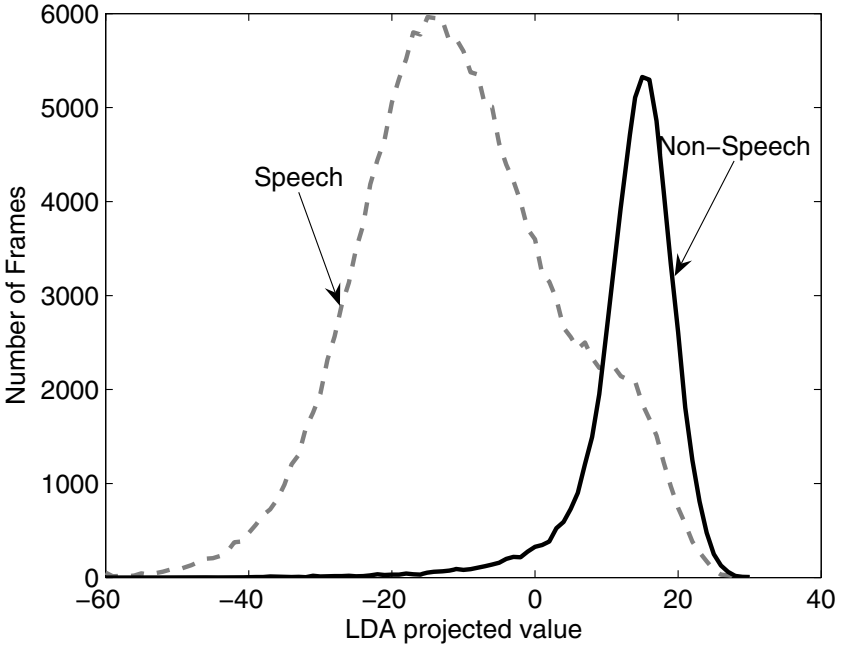


Fig. 1. Histograms of the projected values of speech and non-speech data after LDA for a sample recording

Furthermore, for the training sequences reference files were provided that indicate the speech and non-speech segments.

For the training part the steps of the algorithm are the following:

1. The audio signal is separated into speech and non-speech data.
2. The two segments are separated into overlapping frames.
3. Each frame is multiplied by a Hamming window and then its MFCC are computed.
4. The eigenvector that maximizes the LDA criterion function is employed as the projecting vector of the testing frames.
5. The threshold is found heuristically by exploiting the information from the development data.

For the testing phase the steps of the algorithm are:

1. Each frame is multiplied by a Hamming window and then its MFCC are computed.
2. The MFCCs of each frame are computed.
3. The processed frames are projected in the direction of the eigenvector found in step 4 of the training phase.
4. A decision is made based on a threshold found in step 5 of the training phase.

5. Median filtering is applied to remove abrupt transitions between speech and non-speech regions.

For both the training and the testing phase the frames had a duration of 1024 samples. The amount of overlapping between two neighboring frames was 75%. Finally there were 36 elements in each vector after the MFCC calculation. The first element was the log energy of the frame. The following 11 elements were the MFCC coefficients, and the final 24 elements are the delta and delta-delta coefficients.

3 Speaker Diarization

Diarization is the process of annotating an input audio channel with information that attributes (possibly overlapping) temporal regions of signal energy to their specific sources. These sources can include particular speakers, music, background noises and other signal source/channel characteristics. In our case we are interested in the diarization of the different speakers (whose identities and number are not known a priori). We therefore consider vocal noise such as laughs and coughs to be silence in constructing segment boundaries, as defined in the overall evaluation plan [20]. The segmentation of continuous audio is useful as a preprocessing step for further classification of the segments in applications such as speaker identification/verification, noise rejection, music suppression, etc. This problem has received much attention from researchers and a number of algorithms have been proposed to tackle it, such as a Gaussian Mixture Model and Universal Background Model (GMM-UBM) framework [15], evolutive Hidden Markov Models (E-HMM) [16] and local Gaussian divergence measures [17].

Our system is based on the BIC method for speech segmentation [18], followed by a two-step clustering strategy to group segments from the same speaker together. The final number of clusters (i.e. different speakers) is chosen so that these clusters account (when aggregated) for a percentage of the total number of segments that is higher than a heuristically selected threshold (derived from the RT-05S evaluation data which we used for system development and parameter fine-tuning). Segments that had been assigned to the remaining speakers (if any) are then assigned to one of the surviving speakers based on their a posteriori log likelihoods with respect to the GMMs [19] describing the surviving speakers. Figure 2 illustrates the building blocks of the system (in both the primary and the contrastive case), which are further described in the following subsections.

3.1 Segmentation of Input into Frames

We begin by breaking up the input signal into overlapping frames and extracting 26-D MFCC. Each frame consists of 13 static coefficients (including log energy) and their first-order derivatives (delta coefficients), extracted from 1024 samples of the waveform, with an overlap of 75% between consecutive frames. We did not use delta-delta coefficients in this case, as our experiments with the RT-05S evaluation data indicated a slight performance drop when these extra coefficients were introduced.

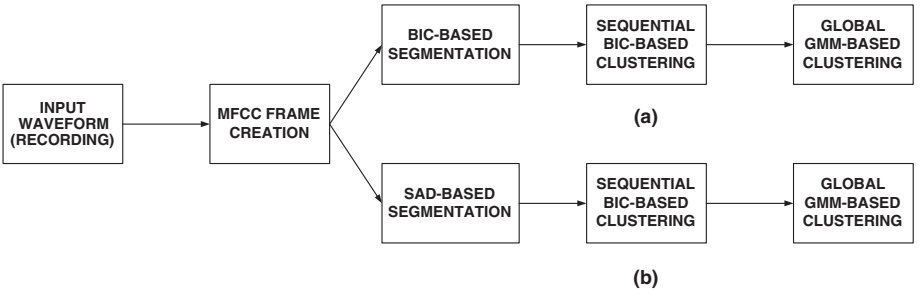


Fig. 2. Block diagram of the (a) primary, and (b) contrastive Speaker Diarization system

3.2 Speech Segmentation Using BIC

For the segmentation step we use the standard BIC algorithm [18], applied to the frames of MFCC. The minimum window size in which a change in speaker is sought based on BIC is dictated by the dimension of these vectors, with the caveat that there must be at least enough vectors to estimate the two covariance matrices of the possible subsegments. If no change is detected in the current window, its size is increased by a fixed step (its starting point remains the same) and the process is repeated until either a change is detected, or a maximum window size is reached. In the former case, the window size is reset to its minimum value with a new starting point at the time the change was detected. In the latter case, the window size is again reset to its minimum value; however, the new starting point is computed by subtracting from the end of the current window 50% of the difference between the maximum and minimum windows. This is done to account for changes that were missed because they were very close to the end point of the maximum window.

3.3 Sequential Segment Clustering Using BIC

For the first step of the clustering process we rely again on BIC to merge consecutive segments. This process goes on until either a maximum number of passes is executed, or a pass results in no consecutive segments being merged. This helps us to have as large and homogeneous segments as possible, in order to facilitate the next step of the clustering process. We have found that this sequential merging step leads to a 20-25% reduction in the number of segments that is presented to the next stage. An alternative approach would be to use tree-based instead of sequential clustering, again based on the BIC values of consecutive frames, however our experiments using this strategy on the RT-05S data indicated that its performance was slightly inferior. Given also its considerably higher execution time, we decided not to adopt the tree-based segment clustering approach.

3.4 Global Segment Clustering Using GMM

First of all, the segments that are derived from the previous step are sorted according to their length, so that we base our GMM's for the speakers on as

many data as possible. For each new segment to be processed, we evaluate its a posteriori log likelihood with respect to all GMM's of currently detected speakers (if any are available). The segment is then assigned to the speaker whose GMM gives the highest average log likelihood over the whole segment only if that log likelihood is higher than a heuristically defined threshold (derived from the RT-05S evaluation data). If this is not the case, we consider this segment to belong to a speaker we have not encountered yet and create a corresponding GMM, provided that the number of speakers found is less than a predefined maximum value (this was set to 10 in the scope of this evaluation, as any meeting was expected to have at most that many participants). If we have reached this threshold though no new speakers will be created, regardless of the log likelihood of the best matching model, and the new segment is assigned to the best matching GMM.

After all segments have been examined in this fashion, we count the votes for each speaker (i.e. the number of segments that have been assigned to that speaker). These votes are then squared, sorted in descending order and their cumulative sum is subsequently computed and divided by the total number of segments. This gives us an indication of the support for each speaker. We keep a number of the most 'popular' speakers so that their cumulative support is less than a heuristically defined threshold (derived from the RT-05S evaluation data).

4 Performance in the RT-06S Evaluation

The RT-06S evaluation plan defines two sub-domains of data: (1) the lecture room and (2) the conference room meetings (lectmtg and confmtg, respectively). The two sub-domains have different sensor setups and different levels of participant interactions, among others. The conference room data were collected from six different sites (1-2 long duration recordings per site), whereas the lecture room data were collected from a total of five sites (28 recordings of around 5 min duration). The total time of conference and lecture data is around 160 and 190 minutes respectively. Each of these sites has different room and microphone configurations. The type of experiment conducted also depends on how many and which of the total microphones were used. The sdm condition uses only a single (usually table-top and in general in the middle of the participants) predefined microphone, whereas the mdm condition allows the use of mupltiple predefined microphones in any manner desired. Each combination of meeting room domain, available microphone inputs and primary or contrastive system (in the case of SPKR), corresponds to a different and unique experiment. Data from previous evaluations were used as training data for the system and to fix a number of threshold parameters. The implemented systems were evaluated both in terms of accuracy (separation of speech and non-speech intervals for SAD, successful segmentation and assignment to the correct speaker for SPKR) and in terms of their speed (as indicated by the ratio of Total Processing Time (TPT) to Source Signal Duration (SSD), denoted as the system's Speed Factor (SF)). For a more detailed discussion of the evaluation specifics see [20].

4.1 Speech Activity Detection

For the SAD task, we have participated only with a primary system for the mdm condition. In this case, the input waveforms for both the training and testing phase have been obtained by performing averaging of all input microphones available based on the evaluation rules. Since these microphones are located throughout the room, this step effectively performs spatial averaging of the channels and suppresses noise by a considerable amount. Our scores in the two meeting scenarios are reported in Table 1, whereas the speed measurements are presented in Table 2. The experiments were executed on a twin-processor Xeon at 2.8 GHz with 2 GB of RAM, running SuSe Linux 9.3.

Table 1. SAD Scores in the RT-06S evaluation

Experiment	confmtg_p_mdm	lectmtg_p_mdm
Score (%)	10.97	13.39
St. Dev. (%)	13.80	15.29
Minimum (%)	1.88	2.71
Maximum (%)	43.61	71.59
Median (%)	7.58	10.59

Table 2. SAD Processing Time Calculation in the RT-06S evaluation

Experiment	confmtg_p_mdm	lectmtg_p_mdm
TPT (sec)	92.44	545.2
SSD (sec)	9723.905	11400.00
SF	0.0095	0.0478

As expected, there was a large standard deviation for both the lecture and conference meeting scenarios. This is due to the non uniform availability of training data for each site. For example, a site that had more training data will give better results than a site with less data available. The system is in any case extremely fast and suitable for any kind of real time application as indicated by the small speed factors. Notice that the discrepancy in speed factors between the lecture and conference meeting scenarios is due to their different sampling rates (44.1 vs. 16 kHz), which means a larger number of frames to be processed in the lectmtg case (by a factor of about 2.75, which is reflected in the respective speed factors).

We are therefore mainly interested in further improving the accuracy of the system, which, albeit sufficient, can be further increased. An ideal scenario for a SAD system would be to have a global training scheme and have its parameters changed adaptively. In our case, the training was changed depending on the site and configuration of the room. In order to address this issue we are using the LDA + MFCC criterion as a part of a collection of features. Our goal is to fuse the different features in a way that will give optimal results in real time.

4.2 Speaker Diarization

For the Speaker Diarization task, we have participated with a primary system for the mdm and sdm conditions, as well as with a contrastive system for the conference meeting mdm experiment. As illustrated in Figure 2, the difference between the two versions of the system lies in the first stage; the primary system segments the input based on BIC, whereas the contrastive one relies on the output of the SAD module described in this paper. For all experiments involving mdm audio conditions, the input waveforms for both the primary and the contrastive system were obtained by performing averaging of all input microphones available based on the evaluation rules. Since these microphones were spread throughout the room, this step effectively performs spatial averaging of the channels. Our scores (taking into account speech overlap regions) in the five experiments are reported in Table 3, whereas the speed measurements are presented in Table 4. The experiments were executed on a twin-processor Xeon at 2.4 GHz with 2 GB of RAM, running Windows and MATLAB 7.04.

Table 3. SPKR Scores in the RT-06S evaluation

Experiment	confmtg_p_mdm	confmtg_c_mdm	confmtg_p_sdm	lectmtg_p_mdm	lectmtg_p_sdm
Score (%)	70.70	66.06	67.22	51.20	45.52
St. Dev. (%)	8.78	9.37	6.23	26.92	19.88
Minimum (%)	62.61	52.87	62.80	19.30	19.35
Maximum (%)	89.86	82.91	77.72	148.15	91.63
Median (%)	69.32	68.55	63.70	53.64	44.60

Table 4. SPKR Processing Time Calculation in the RT-06S evaluation

Experiment	confmtg_p_mdm	confmtg_c_mdm	confmtg_p_sdm	lectmtg_p_mdm	lectmtg_p_sdm
TP (sec)	863.37	930.34	849.52	2344.54	2346.90
SSD (sec)	9723.905	9723.905	9723.905	11400.00	11400.00
SF	0.0888	0.0957	0.0874	0.2057	0.2059

The first comment to be made on the results of Table 3 is that the system performs on average 20% better (depending on microphone conditions) in the lectmtg scenario. This is to be expected since there is considerably less interaction in those recordings. It is therefore much easier to perform diarization, since most of the speech comes from a single speaker (i.e., the presenter). What is also interesting is that the contrastive system appears to perform slightly better than the primary one, which means that we can take advantage of a good SAD module to enhance the performance of segmentation. The reason behind this seems to be that by pre-processing frames with a SAD module we have a more accurate indication of when speaker changes occur, since each silence interval corresponds to a speaker boundary. However, despite the beneficial effect of the SAD module, the system is far from perfect, especially in cases where there is

intense dialogue and speakers take short turns as in the confmtg case. We therefore aim to consider other approaches to improve the accuracy of our system, as well as to be able to handle intervals of concurrent speech from two or more speakers, which are not currently supported by our algorithm.

Regarding the speed of the implemented SPKR module, we should note that the current version is not real-time, due mainly to the fact that it is written in MATLAB. We expect that porting the system to C++ should provide enough of an execution time reduction to allow it to be used in a real-time application.

5 Conclusion

In this paper we have discussed the SAD and SPKR systems that were developed by the Athens Information Technology in the scope of the NIST RT-06S evaluations. The SAD module is based on LDA to perform classification of recorded frames into speech and non-speech, whereas the SPKR one segments recordings into speech intervals based on the Bayesian Information Criterion, and then assigns them to a set of speakers (with size and composition that are a priori unknown) following a two-step clustering strategy to group segments from the same speaker together. We have then proceeded to report the performance (in terms of both accuracy and speed) of these subsystems in the NIST RT-06S evaluation, as well as outline a number of improvements that we are considering to counter each module's deficiencies.

References

1. Weiser, M.: The Computer for the 21st Century. Scientific American, vol. 265, no. 3 (1991) 66-75
2. Waibel, A., Steusloff H., Stiefelhagen, R., et. al: CHIL: Computers in the Human Interaction Loop. 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Lisbon, Portugal, April 2004.
3. Pnevmatikakis, A., Talantzis, F., Soldatos, J., Polymenakos, L.: Robust Multimodal Audio-Visual Processing for Advanced Context Awareness in Smart Spaces I. Maglogiannis, K. Karpouzis and M. Bramer (eds.), Artificial Intelligence Applications and Innovations (AIAI06), Springer, Berlin Heidelberg, pp. 290-301, June 2006.
4. http://www.clear-evaluation.org/
5. Katsarakis, N., Souretis, G., Talantzis, F., Pnevmatikakis, A., Polymenakos, L.: 3D Audiovisual Person Tracking Using Kalman Filtering and Information Theory. R. Stiefelhagen and J. Garofolo (eds.): CLEAR 2006, Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, 2006.
6. Stergiou, A., Pnevmatikakis, A., Polymenakos, L.: A Decision Fusion System across Time and Classifiers for Audio-visual Person Identification. R. Stiefelhagen and J. Garofolo (eds.): CLEAR 2006, Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, 2006.
7. Stergiou, A., Pnevmatikakis, A., Polymenakos, L.: Enhancing the Performance of a GMM-based Speaker Identification System in a Multi-Microphone Setup. Accepted, INTERSPEECH 2006, Pittsburgh, September 2006.

8. Rabiner, L. R., Sambur M. R.: An algorithm for determining the endpoints of isolated utterances. The Bell System Technical Journal, **54** (1975) 297
9. Li, K., Swamy, N. S., Ahmad, M. O.: An Improved Voice Activity Detection Using Higher Order Statistics. IEEE Transactions on Speech and Audio Processing, Vol. 13, NO. 5, September 2005
10. Stegmann, J., Schroeder, G.: Robust Voice Activity Detection Based on the Wavelet Transform. Proc. IEEE Workshop on Speech Coding For Telecommunications, Pocono Manor, Pennsylvania, USA, pp. 99-100, September 1997
11. Reynolds, D. A., Rose, R. C., Smith, M. J. T.: PC-Based TMS320C30 Implementation of the Gaussian Mixture Model Text-Independent Speaker Recognition System. International Conference on Signal Processing Applications and Technology, Hyatt Regency, Cambridge, Massachusetts, pp. 967-973, November 1992
12. Martin, A., Charlet, C., Mauary, L.: Robust Speech/Non- Speech Detection Using LDA Applied to MFCC IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, 2001
13. Duda, R., Hart, R., Stork, D.: Pattern Classification. Wiley-Interscience, New York, 2001.
14. Rabiner, L., Schafer, R.: Digital Processing of Speech Signals. Prentice Hall Series in Signal Processing, September 1978.
15. Wu, T.-Y., Lu, L., Chen, K., Zhang, H.-J.: Universal Background Models for Real-Time Speaker Change Detection MMM 2003, 135-149
16. Moraru, D., Meignier, S., Fredouille, C., Besacier, L., Bonastre, J.-F.: The ELISA consortium approaches in broadcast news speaker segmentation during the NIST 2003 rich transcription evaluation. Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP 2004), Montreal, Canada, 2004.
17. Gauvain, J.L., Lamel L., Adda, G.: Partitioning and transcription of broadcast news data. In International Conference on Speech and Language Processing, volume 4, pages 1335-1338, Sydney, Australia, Dec 1998.
18. Tritschler, A., Gopinath, R.: Improved speaker segmentation and segments clustering using the Bayesian Information Criterion. Proc. of Eurospeech, pp. 679-682, 1999.
19. Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Transactions on Speech and Audio Processing, Vol. 3, No. 1, pp. 72 - 83, January 1995.
20. Fiscus, J.: Spring 2006 (RT-06S) Rich Transcription Meeting Recognition Evaluation Plan (v2) [http://www.nist.gov/speech/tests/rt/rt2006/spring/docs/rt06s-meeting-eval-plan-V2.pdf].

Speaker Diarization: From Broadcast News to Lectures

X. Zhu, C. Barras, L. Lamel, and J-L. Gauvain*

LIMSI-CNRS, BP 133
91403 Orsay Cedex, France

Abstract. This paper presents the LIMSI speaker diarization system for lecture data, in the framework of the Rich Transcription 2006 Spring (RT-06S) meeting recognition evaluation. This system builds upon the baseline diarization system designed for broadcast news data. The baseline system combines agglomerative clustering based on Bayesian information criterion with a second clustering using state-of-the-art speaker identification techniques. In the RT-04F evaluation, the baseline system provided an overall diarization error of 8.5% on broadcast news data. However since it has a high missed speech error rate on lecture data, a different speech activity detection approach based on the log-likelihood ratio between the speech and non-speech models trained on the seminar data was explored. The new speaker diarization system integrating this module provides an overall diarization error of 20.2% on the RT-06S Multiple Distant Microphone (MDM) data.

1 Introduction

Audio diarization is the process of partitioning an input audio stream into homogeneous segments according to their specific audio source such as speaker identity, category of music, background noise or channel conditions. Speech activity detection (SAD) is the simplest case of diarization, which just divides the audio data into speech/non-speech segments. Speaker diarization, also referred to as speaker segmentation and clustering, is a more complicated task than audio diarization, and needs to determine segments consisting of the speech from only one speaker and associate speech segments from the same speaker. SAD is a very useful preprocessing step for many audio technologies such as automatic speech recognition, speaker identification and verification, speaker localization etc. Speaker diarization has been used in Automatic Speech Recognition (ASR) to carry out unsupervised speaker adaptation, where the amount of data available for the adaptation can be increased by clustering segments from the same speaker. Speaker diarization can also improve the readability of an automatic transcription by structuring the audio stream into speaker turns and is of interest for the indexing of multimedia documents.

The challenges for the speaker diarization task derive from the varied data types: Broadcast News (BN), telephone conversations and meeting recordings. Most research efforts on speaker diarization have focused on the broadcast news domain [1,2]. Recently there has been strong interest in the meeting domain [3,4], which poses more

* This work was partially financed by the European Commission under the FP6 Integrated Project IP 506909 CHIL.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 396–406, 2006.

difficulties for the speaker diarization task. The speech in the meeting is completely spontaneous, with frequent periods of overlapping speech and a large number of silence segments for any given speaker. Meetings are usually recorded using different types of microphones located at various positions in the room, providing multiple audio files with different signal qualities for the same meeting. The use of the distant microphones also makes the audio signal more noisy than many of the broadcast news recordings.

In the Rich Transcription 2006 Spring (RT-06S) meeting recognition evaluation [5], the task was divided into two sub-domains: conference room meetings and lecture room meetings (seminar-like meetings). Compared with the conference data, the lecture meetings have less interaction between the participants, and typically consist of a presentation from a lecturer followed by a question/answering session or discussion period.

LIMSI participated in the speech activity detection and speaker diarization tasks of the RT-06S evaluation, focusing on the lecture data. The LIMSI multi-stage speaker diarization system developed for BN data [6] was adapted to the lecture data, especially the SAD module. This modified system was tested on far-field conditions: the Multiple Distant Microphone (MDM), Single distant Microphone (SDM) and Multiple Mark III Microphone Array (MM3A). As defined by this evaluation, no a priori knowledge of the speaker's voice or even the number of speakers is available for the distant microphone conditions, and thus only a relative and recording-internal speaker identification is produced by the system.

The remainder of this paper is organized as follows: Section 2 describes the baseline speaker diarization system for broadcast news data, and Section 3 presents the log-likelihood based speech activity detection adapted to the lecture data. The experimental results are presented in Section 4, followed by some conclusions.

2 Baseline BN Diarization System

The baseline speaker diarization system developed for Broadcast News combines an agglomerative clustering based on Bayesian information criterion (BIC) with a second clustering stage which uses state-of-the-art speaker identification (SID) methods. It obtains good performance on BN data, achieving an overall speaker diarization error of 8.5% on RT-04F evaluation data [7]. The primary system is structured as follows:

2.1 Feature Extraction

Mel frequency cepstral parameters are extracted from the speech signal every 10 ms using a 30 ms window on a 0-8kHz band. The 38 dimensional feature vector consists of 12 cepstral coefficients, Δ and Δ-Δ coefficients plus the Δ and Δ-Δ log-energy. Acoustic vector normalization is only performed in the SID clustering stage.

2.2 Speech Activity Detection (SAD)

Speech is extracted from the signal with a Viterbi decoding using Gaussian Mixture Models (GMM) for speech, noisy speech, speech over music, pure music, and silence or noise. The aim of the SAD is to remove only long regions without speech such as silence, music and noise, so the penalty of switching between models in the Viterbi

decoding was set to minimize the loss of speech signal. The GMMs, each with 64 Gaussians, were trained on about 1 hour of the specific type of data, selected from English Broadcast News data distributed by the Linguistic Data Consortium (LDC).

2.3 Initial Segmentation

The segmentation process consists of finding segment boundaries that correspond to the instantaneous speaker change points. The initial segmentation of the signal is performed by taking the maxima of a local Gaussian divergence measure between two adjacent sliding windows s_1 and s_2 of 5 seconds, similar to the KL2 metric based segmentation [8]. Each window is modeled by a single diagonal Gaussian using the static features (i.e., only the 12 cepstral coefficients plus the energy). More precisely, the Gaussian divergence measure is defined as:

$$G(s_1, s_2) = (\mu_2 - \mu_1)' \Sigma_1^{-1/2} \Sigma_2^{-1/2} (\mu_2 - \mu_1) \quad (1)$$

with $s_i \sim \mathcal{N}(\mu_i, \Sigma_i)$ and Σ_i diagonal, $i \in \{1, 2\}$. The detection threshold was optimized on the training data in order to provide acoustically homogeneous segments.

2.4 Viterbi Resegmentation

An 8-component GMM with a diagonal covariance matrix is trained on each segment resulting from the initial segmentation, the boundaries of the speech segments detected by the SAD module are then refined using a Viterbi segmentation with this set of GMMs.

2.5 BIC Clustering

An initial cluster c_i is modeled by a single Gaussian with a full covariance matrix Σ_i estimated on the n_i acoustic frames of each segment output by the Viterbi resegmentation. The BIC criterion [9] is used both for the inter-cluster distance measure and the stop criterion. It is defined as:

$$\Delta BIC = (n_i + n_j) \log |\Sigma| - n_i \log |\Sigma_i| - n_j \log |\Sigma_j| - \lambda \frac{1}{2} (d + \frac{1}{2} d(d+1)) \log n \quad (2)$$

where d is the dimension of the feature vector space, $n = n_i + n_j$ and λ weights the BIC penalty. At each step the two nearest clusters are merged, and the ΔBIC values between this new cluster and the remaining clusters are computed. This clustering procedure stops when all ΔBIC are greater than zero.

2.6 SID Clustering

After the BIC clustering stage, speaker recognition methods [11,12] are used to improve the quality of the speaker clustering. Feature warping normalization [13] is performed on each segment using a sliding window of 3 seconds in order to map the cepstral feature distribution to a normal distribution and reduce the non-stationary effects of the acoustic environment. The GMM of each remaining cluster is obtained by maximum a

posteriori (MAP) adaptation [15] of the means of the matching Universal Background Model (UBM [14]). For each gender and channel condition (wide band, narrow band) combination, an UBM with 128 diagonal Gaussians is trained on the corresponding subset of 1996/1997 English Broadcast News data. Then a second stage of agglomerative clustering is performed using the cross log-likelihood ratio as in [16]:

$$\mathcal{S}(c_i, c_j) = \frac{1}{n_i} log \frac{f(x_i|M_j)}{f(x_i|B)} + \frac{1}{n_j} log \frac{f(x_j|M_i)}{f(x_j|B)} \quad (3)$$

where $f(\cdot|M)$ is the likelihood of the acoustic frames given the model M, and n_i is the number of frames in cluster c_i. The clustering stops when the cross log-likelihood ratio between all clusters is below a given threshold δ optimized on the development data (see Section 4.1).

2.7 SAD Post-filtering

The word segmentation output by the LIMSI Broadcast News Speech-To-Text system [17] is used in a post-processing stage to filter out short-duration silence segments that were not removed by the initial speech activity detection step. Only inter-word silences longer than 1 second are filtered out, this value having been determined empirically.

3 Speech Activity Detection for Lectures

The LIMSI RT-06S speaker diarization system for lecture data was built upon the broadcast news diarization system. Initial results on the development data with the baseline system had a high speech activity detection error, especially with a lot of missed speech, therefore a different approach for SAD was explored. One weakness of the standard Viterbi decoding is the lack of temporal control for each model. A transition penalty can be used to control the size of the segments, but as the level of noise increases, the likelihood of the speech model will decrease and thus the shortest speech segments will be discarded. Instead of setting a minimal likelihood level for switching from one model to the other, it is easier to choose a minimal duration for speech and non-speech segments.

We designed a simple speech activity detector based on the log-likelihood ratio (LLR) between the speech and non-speech models, and replaced the Viterbi decoding with a simple smoothing of the LLR followed by a decision module. More precisely:

- for each frame x_i, the LLR r_i between the speech and non-speech models λ_S and λ_N is computed taking into account their prior probabilities $P(S)$ and $P(N)$:

$$r_i = \log f(x_i|\lambda_S)P(S) - \log f(x_i|\lambda_N)P(N)$$

- two adjacent smoothing windows with a duration of $w = 100$ frames (i.e. 1 second) sliding over the signal are used for the detection of speech and non-speech transitions. A transition is possible when the sign of the averaged LLR in the left and right windows changes around the current frame:

$$s_i^+ . s_i^- < 0 \quad \text{with} \quad s_i^+ = \frac{1}{w} \sum_{j=i+1}^{i+w} r_j \quad \text{and} \quad s_i^- = \frac{1}{w} \sum_{j=i-w}^{i-1} r_j$$

– for a set I consisting of contiguous candidate transitions, the position of the transition is chosen at the maximum of difference between the averaged ratio of the left and right windows:

$$i^* = \underset{i \in I}{\operatorname{argmax}} |s_i^+ - s_i^-|$$

The GMMs for speech and non-speech were trained on about 2 hours of far-field data from seminars recorded at the University of Karlsruhe (UKA).

4 Experimental Results

In RT-06S lecture room evaluation, results were submitted for the SAD and speaker diarization tasks on three audio input conditions: MDM, SDM and MM3A. The configurations of BIC clustering and SID clustering were optimized on the development data. All experiments were carried out with the BIC penalty weight $\lambda = 3.5$ and the SID threshold $\delta = 0.5$.

4.1 Performance Measures and Databases Descriptions

The speaker diarization task performance is measured via an optimum mapping between the reference speaker IDs and the hypotheses. This is the same metric as was used to evaluate the performance on BN data. The primary metric for the task, referred to as the speaker error, is the fraction of speaker time that is not attributed to the correct speaker, given the optimum speaker mapping. In addition to this speaker error, the overall speaker diarization error rate (DER) also includes the missed and false alarm speaker times. The SAD task performance is evaluated by summing the missed and false alarm speaker error. In RT-06S evaluation, the metrics are calculated over all the speech, including the overlapping speech. The DER restricted to non-overlapping speech segments is also given for comparison purposes.

The experiments were conducted on the NIST RT-06S evaluation data comprised of lectures provided by the CHIL (Computer in the Human Interaction Loop) consortium. The data were collected at 5 of the CHIL partner sites: AIT (Athens Information Technology), IBM, ITC (Istituto Trentino di cultura), UKA and UPC (Universitat Politecnica de Catalunya). The development dataset (dev) consists of all seminars used as RT-05s evaluation data, plus an additional seminar from UKA and four seminars from AIT, IBM, ITC and UPC one each. The evaluation dataset (eval) is composed of 38 seminar segments each lasting about 5 minutes.

4.2 Audio Input Selection

For the MDM evaluation condition, a single microphone signal randomly chosen from the available MDM channels and different from the channel selected for the SDM condition was used as the input to the speaker diarization system. Because the same microphone type is used for the MDM and SDM conditions, no individual development was carried on SDM condition, i.e. the same configuration for the speaker diarization system is adopted for both conditions. The microphone channels used for the MDM and SDM conditions are detailed in Table 1. For MM3A evaluation condition, the beamformed multiple mark III microphone array data provided by UKA was used as the input of the speaker diarization system.

Table 1. Channel selection for the MDM and the SDM conditions for the dev and eval data

Dataset	*Condition*	*AIT*	*IBM*	*ITC*	*UKA*	*UPC*
dev	MDM	mic05	Audio_17	Table-1	TableTop-1	channel15
eval	MDM	mic06	Audio_17	Table-2	TableTop-1	channel16
eval	SDM	mic05	Audio_19	Table-1	Table-2	channel15

4.3 RT-06S MDM Development Results

The performances of the speaker diarization systems integrating different SAD modules are summarized in Table 2. The "vit-bn" system uses Viterbi decoding with 5 GMMs (64 Gaussians) for speech, noisy speech, speech over music, pure music, and silence, each trained on one hour of BN data. This baseline speaker diarization system is the same system as was used in RT-04F evaluation for BN data. The "vit-bn+mt" system uses Viterbi decoding with GMMs trained on the BN data plus 2 GMMs (256 Gaussians) for speech and non-speech trained on 2 hours of far-field data from the UKA seminars. The "vit-mt" system uses Viterbi decoding only with speech and non-speech models trained on lecture data. The "slr-mt" system uses the smoothed LLR-based SAD method with a prior probability of 0.2 for the non-speech model and 0.8 for the speech model. As can be seen in Table 2, Viterbi SAD using the models trained on both BN and lecture data have very high missed speech error rates (ranging from 18% to 14%) on the MDM development data. The log-likelihood based SAD substantially reduces this error (2.7% missed speech error) with limited increase in false alarm speech error. Compared with the baseline speaker diarization system, a relative DER reduction of 33% is obtained by the system using the smoothed LLR-based SAD.

Table 2. Speaker diarization errors on the MDM development data for different SAD modules

System	*Missed speech (%)*	*False alarm speech (%)*	*Speaker error (%)*	*Overlap DER (%)*
vit-bn (baseline)	18.2	3.0	9.0	30.2
vit-bn+mt	19.3	2.9	8.7	31.0
vit-mt	14.2	3.7	12.4	30.2
slr-mt	**2.7**	**6.1**	**11.7**	**20.5**

Table 3 gives the speaker diarization results on the MDM development data when the number of Gaussians for the speech and the non-speech models used in the smoothed LLR-based SAD are varied. These results are obtained with a prior probability of 0.4 for the non-speech model and 0.6 for the speech model. There are no gains of the overall diarization error when the number of Gaussians is increased from 256 to 512 on the MDM development data.

The effect of the prior probabilities for speech and non-speech used in LLR-based SAD was also studied. The results presented in Table 4 are obtained with 256-component GMMs used for each model. Because it is important for automatic speech transcription to reject the least amount of speech as possible, a higher prior probability

Table 3. Results varying the number of Gaussians for the speech and non-speech models on the MDM development data

nb. Gaussians	*Missed speech (%)*	*False alarm speech (%)*	*Speaker error (%)*	*Overlap DER (%)*
64	9.5	4.0	11.0	24
128	9.5	3.7	11.0	24
256	**7.8**	**4.2**	**11.0**	**23**
512	7.7	4.2	11.1	23

Table 4. Results obtained by using different prior probabilities for the speech and non-speech models on the MDM development data

P(N):P(S)	*Missed speech (%)*	*False alarm speech (%)*	*Speaker error (%)*	*Overlap DER (%)*
0.1:0.9	1.0	9.5	12.0	22.4
0.2:0.8	**2.7**	**6.1**	**11.7**	**20.5**
0.3:0.7	5.2	5.0	11.3	21.5
0.4:0.6	7.8	4.2	11.0	23.0

for the speech model is preferred relative to the non-speech model. As shown in Table 4, using a prior probability of 0.2 for the non-speech model and 0.8 for the speech model provides the best results for both speech activity detection (8.8% SAD error) and speaker diarization (20.5% DER).

After the experiments on the MDM development data, the configuration of the log-likelihood based SAD system is optimized as: a prior probability of 0.2 for the non-speech model and 0.8 for the speech model with 256-component GMMs used for both models. The performance of the speaker diarization system using the LLR-based SAD module is presented in Table 5, where the result is given for the individual seminar having the corresponding reference released by NIST. As shown in Table 5, the average DER of 20.5% masks the large variation across seminars. Normally lower overall diarization error can be obtained on the seminars with only one speaker, but for "UKA_20041124_A_Segment2" seminar, a very high false alarm speech error of about 150% is produced by the LLR-based SAD module. After listening to the audio file, we found that many speech segments are missing in the reference transcription, this may be because the speech signal was not recorded on the microphone channel chosen for the manual reference transcription.

In order to analyze the variation in system performance, we calculated the ratio between the speech time from the main speaker (who spoke the most in the seminar) and the total seminar duration on all the seminars in Table 5 except the "UKA_20041124_A-_Segment2" seminar. Figure 4.3 shows that the speaker diarization system provides lower overall diarization error on seminars where the main speaker spoke for more than 80% of the seminar duration. Moreover a correlation between the DER and the dominant speaker duration ratio is apparent clearly; consistent with the observations reported in [18].

Table 5. Results by seminar in the MDM development dataset, "REF" represents the number of speakers in the reference transcriptions

Seminar	*Missed speech (%)*	*False alarm speech (%)*	*Speaker error (%)*	*Overlap DER (%)*	REF
AIT_20050726_Segment1	0.9	11.3	10.7	22.9	4
IBM_20050824_Segment1	2.6	0.9	1.6	5.1	2
ITC_20050429_Segment1	2.3	4.1	8.1	14.6	3
UKA_20041123_A_Segment1	0.0	0.9	0.0	0.9	1
UKA_20041123_A_Segment2	0.9	0.0	29.6	30.6	2
UKA_20041123_B_Segment2	7.2	35.0	4.6	46.7	3
UKA_20041123_C_Segment1	0.6	1.6	0.0	2.2	1
UKA_20041123_C_Segment2	1.7	5.7	4.1	11.4	3
UKA_20041123_D_Segment1	7.9	1.3	0.8	10.1	1
UKA_20041123_D_Segment2	1.6	75.5	4.8	81.9	2
UKA_20041123_E_Segment1	1.4	0.8	7.6	9.8	2
UKA_20041123_E_Segment2	3.5	5.1	6.6	15.2	2
UKA_20041124_A_Segment1	1.7	7.6	0.1	9.4	1
UKA_20041124_A_Segment2	3.2	149.9	4.5	157.6	1
UKA_20041124_B_Segment1	0.2	1.4	0.3	1.8	1
UKA_20041124_B_Segment2	1.9	3.2	44.2	49.3	4
UKA_20050112_Segment1	4.5	0.3	0.0	4.8	1
UKA_20050112_Segment2	10.2	1.2	7.2	18.7	3
UKA_20050126_Segment1	0.3	2.9	0.0	3.2	1
UKA_20050127_Segment1	1.3	0.2	0.1	1.6	1
UKA_20050128_Segment1	2.3	1.1	0.0	3.5	1
UKA_20050128_Segment2	3.0	1.7	39.5	44.1	5
UKA_20050202_Segment2	8.8	11.0	57.4	77.2	7
UKA_20050209_Segment1	2.5	1.4	0.0	3.9	1
UKA_20050209_Segment2	11.4	11.8	52.7	75.9	4
UKA_20050310_A_Segment1	0.5	1.7	0.8	3.0	1
UKA_20050310_A_Segment2	1.0	4.1	53.9	59.0	4
UKA_20050310_B_Segment1	0.2	0.6	0.0	0.8	1
UKA_20050314_Segment1	3.1	1.8	0.5	5.4	1
UKA_20050314_Segment2	6.0	3.3	19.0	28.3	4
UPC_20050601_Segment1	2.7	24.4	20.0	47.1	3
all	2.7	6.1	11.7	20.5	-

4.4 RT-06S Evaluation Results

The RT-06S evaluation results are given in Table 6. For the MDM and SDM conditions, system tuning used the same development data, and therefore identical configurations are used for both conditions. The system performance is quite similar to that obtained on the MDM development data with an overlap overall diarization error of 21.5%. For the SDM audio input condition, the overlap DER is increased to 24.5%. This increase of the diarization error comes mainly from the SAD error, due to the different quality of the microphone channels used for the MDM and SDM conditions.

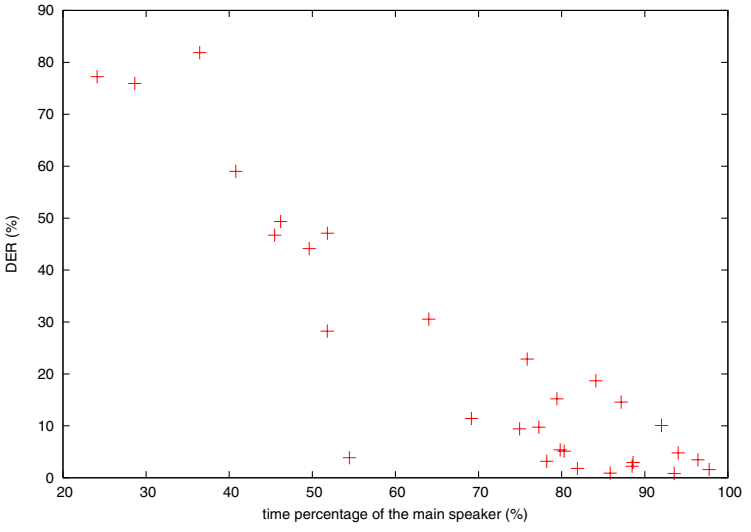


Fig. 1. Overall speaker diarization error on the MDM development data as a function of the time percentage of speech for the main speaker during the seminar

Table 6. Evaluation results for SAD and speaker diarization for the MDM, SDM and MM3A conditions

Condition	*nb. Gaussians*	*P(S)*	*Overlap SAD error (%)*	*Overlap DER (%)*	*Non-overlap DER (%)*
MDM	256	0.8	9.0	21.5	20.2
SDM	256	0.8	12.4	24.5	23.2
MM3A	128	0.6	11.5	25.9	24.7

For the MM3A contrast condition, the system configuration was optimized on the beamformed development data. Since no adaptation of the SAD acoustic models is performed on the beamformed data, a slightly higher diarization error of 25.9% is obtained for the MM3A condition relative to the MDM condition.

5 Conclusions

The work at LIMSI related to speech activity detection and speaker diarization in the framework of the RT-06S meeting recognition evaluation was reported in this paper. Our speaker diarization system for the lecture task builds upon a baseline multi-stage system developed for broadcast news. The main modification is the use of a smoothed log-likelihood ratio based SAD with acoustic models adapted to the lecture data. This SAD was demonstrated to perform much better than the baseline Viterbi SAD. On the MDM development data, the LLR-based SAD provides a significant reduction of the SAD error up to 58% relative to Viterbi SAD, and in particular reduces the missed

speech error. Concerning the speaker diarization performance, the diarization system using the LLR-based SAD gives an overall error of 20% , compared to the 30% overall error obtained with the baseline system. On the evaluation data, the RT-06S speaker diarization system provides an overlap overall diarization error of 21.5% on the MDM condition, with a small increase in the overlap DER to 24.5% for the SDM condition and a higher error of 25.9% for the MM3A condition. The robustness of the speaker diarization system depends a lot on the data domain. The combination of BIC clustering and SID clustering is very effective on the BN data and provides 8.5% non-overlapping overall diarization error on RT-04F evaluation data. A relatively higher non-overlapping DER of 20.2% is obtaind on the MDM lecture data. This decrease of the speaker diarization performance may derive from the lower signal quality of the lecture data.

Our future work will focus on the improvement of the robustness of the speaker diarization system. Efficiently using information from all of the available MDM microphone channels is another important research direction.

References

1. S. E. Tranter and D. A. Reynolds, "Speaker diarisation for broadcast news," in *Proc. ISCA Speaker Recognition Workshop Odyssey 2004*, Toledo, Spain, May 2004.
2. C. Barras, X. Zhu, S. Meignier and J-L. Gauvain, "Multi-Stage Speaker Diarization of Broadcast News," to appear in *The IEEE Transactions on Audio, Speech and Language Processing*, September, 2006 (to appear).
3. X. Anguera, C. Wooters, B. Peskin and M. Aguilo, "Robust Speaker Segmentation for Meeting: The ICSI-SRI Spring 2005 Diarization System," in *MLMI 2005 Meeting Recognition Workshop*, Edinburgh, UK, July 2005.
4. D. Istrate, C. Fredouille, S. Meignier, L. Besacier and J-F. Bonastre, "NIST RT05S evaluation: pre-processing techniques and speaker diarization on multiple microphone meetings," in *MLMI 2005 Meeting Recognition Workshop*, Edinburgh, UK, July 2005.
5. NIST, "Spring 2006 Rich Transcription (RT-06S) Meeting Recognition Evaluation Plan," `http://www.nist.gov/speech/tests/rt/rt2006/spring/docs/rt06s-meeting-eval-plan-V2.pdf`, February, 2006.
6. X. Zhu, C. Barras, S. Meignier, and J-L. Gauvain, "Combining Speaker Identification and BIC for Speaker Diarization," in *ISCA Interspeech'05*, Lisbon, September 2005, pp. 2441–2444.
7. NIST, "Fall 2004 Rich Transcription (RT-04F) evaluation plan," `http://www.nist.gov/speech/tests/rt/rt2004/fall/docs/rt04f-eval-plan-v14.pdf`, August 2004.
8. M. Siegler, U. Jain, B. Raj, and R. Stern, "Automatic segmentation and clustering of broadcast news audio," in *the DARPA Speech Recognition Workshop*, Chantilly, USA, Feb. 1997.
9. S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *DARPA Broadcast News Transcription and Understanding Workshop*, Landsdowne, USA, Feb. 1998.
10. M. Cettolo, "Segmentation, classification and clustering of an Italian broadcast news corpus," in *Conf. on Content-Based Multimedia Information Access (RIAO 2000)*, Paris, April 2000.
11. J. Schroeder and J. Campbell, Eds., *Digital Signal Processing (DSP), a review journal - Special issue on NIST 1999 speaker recognition workshop*, Academic Press, 2000.
12. C. Barras and J.-L. Gauvain, "Feature and score normalization for speaker verification of cellular data," in *IEEE ICASSP 2003*, Hong Kong, 2003.

13. J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. ISCA Speaker Recognition Workshop Odyssey 2001*, Chania, Crete, June 2001, pp. 213–218.
14. D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing (DSP), a review journal - Special issue on NIST 1999 speaker recognition workshop*, vol. 10, no. 1-3, pp. 19–41, 2000.
15. J.-L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2(2), pp. 291–298, April 1994.
16. D. A. Reynolds, E. Singer, B. A. Carlson, G. C. O'Leary, J. J. McLaughlin, and M. A. Zissman, "Blind clustering of speech utterances based on speaker and language characteristics," in *Proc. of International Conf. on Spoken Language Processing (ICSLP'98)*, 1998.
17. L. Nguyen, S. Abdou, M. Afify, J. Makhoul, S. Matsoukas, R. Schwartz, B. Xiang, L. Lamel, J.-L. Gauvain, G. Adda, H. Schwenk, and F. Lefevre, "The 2004 BBN/LIMSI 10xRT English broadcast news transcription system." in *DARPA RT04'S*, Palisades, NY, Nov 2004.
18. N. Mirghafori and C. Wooters, "Nuts and Flakes: A Study of Data Characteristics in Speaker Diarization," in *IEEE ICASSP 2006*, pp. 1017–1020, Toulouse, May 2006.

The ISL RT-06S Speech-to-Text System

Christian Fügen[1], Shajith Ikbal[1], Florian Kraft[1], Kenichi Kumatani[1], Kornel Laskowski[1], John W. McDonough[1], Mari Ostendorf[1,2], Sebastian Stüker[1], and Matthias Wölfel[1]

[1] Interactive Systems Laboratories, Universität Karlsruhe (TH), Karlsruhe, Germany
[2] Dept. of Electrical Engineering, University of Washington, Seattle, WA, USA
{fuegen,shajith,fkraft,kumatani,kornel,jmcd,mo,stueker,wolfel}@ira.uka.de

Abstract. This paper describes the 2006 lecture and conference meeting speech-to-text system developed at the Interactive Systems Laboratories (ISL), for the individual head-mounted microphone (IHM), single distant microphone (SDM), and multiple distant microphone (MDM) conditions, which was evaluated in the RT-06S Rich Transcription Meeting Evaluation sponsored by the US National Institute of Standards and Technologies (NIST). We describe the principal differences between our current system and those submitted in previous years, namely improved acoustic and language models, cross adaptation between systems with different front-ends and phoneme sets, and the use of various automatic speech segmentation algorithms.

1 Introduction

In this paper, we present the ISL's most recent speech-to-text systems for lectures and conference meetings, which have evolved significantly over previous versions [1,2,3] and which were evaluated in the NIST RT-06S Rich Transcription Meeting Evaluation.

The systems described in [1] and [3] shared many common elements, e.g. front-end, phoneme set, and training strategy. The systems described in this paper differ from them in several important ways. Notably, we used only speaker-adapted acoustic models. Even in the first pass, we used models trained with vocal tract length normalization (VTLN), and employed speaker-based incremental adaptation during decoding. Several acoustic models with different front-ends were trained: besides our standard FFT MFCC front-end, we also trained a system with a minimum variance distortionless response (MVDR) [4] front-end. Furthermore, in addition to our standard phoneme set, which was used in RT-04S [3], we also trained a system based on the PRONLEX phoneme set in order to exploit benefits from cross-adaptation and system combination [5]. We also improved our language models by incorporating data collected from the world wide web. Last but not least, we used different speech segmentation algorithms compared to the one used in our RT-04S evaluation system [6].

Most of the decoding experiments described in this paper were conducted on the lecture meeting portion of the official RT-06S development set, which is

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 407–418, 2006.

identical to the RT-05S evaluation set further referred to as *lectDEV*, and only a small portion of experiments were done on the conference meeting portion of this set, *confDEV*. The corresponding evaluation sets are named *lectEVAL* and *confEVAL*. The *confDEV* results in this paper exclude one NIST meeting (`NIST_20050412-1303`), as it contained a participant on speakerphone, a condition which was guaranteed not to appear in *confEVAL*.

2 Automatic Segmentation

Automatic segmentation for the various conditions of the lecture and conference subtasks is provided by different systems. For the IHM condition, which is particularly difficult due to cross-talk from background speakers, we developed separate systems for the lecture and conference meeting subtasks. The lecture segmenter is an improved version of a single-microphone system which we used in the TC-STAR project [7]; the conference segmenter is a further evolution of our multi-microphone RT-04S IHM segmentation system. During RT-06S development, the two IHM systems further diverged due to subdomain-specific challenges: participants without microphones in the lecture task, and overlap and participant interaction in the conference task. We describe the two IHM segmenters below. The MDM and SDM segmentation was the same for both the lecture and conference meeting subtasks, and brief mention is included in the lecture segmentation description.

Lecture Meeting Segmentation
Our IHM lecture segmentation approach uses the following speech activity features extracted with a frame size of 32 ms and a frame shift of 10 ms: frame energy in decibels (E), mean and variance normalized E passed through a sigmoid function (E_n), energy-normalized linear prediction error (L) [8], slope along the frequency axis of a mel-warped filter-bank spectrum (S), and SPEECH/NON-SPEECH posteriors (P) computed using a multi-layer perceptron (MLP) trained with standard MFCC features. Segmentation for the IHM condition is performed in three steps [9]:

1. *Background speech activity rejection* discards regions of prominent cross-talk. This step uses E from all available microphones for a particular meeting and additional constraints such as presence of a minimal percentage of voiced speech and minimum duration.
2. *Foreground speech activity detection* identifies regions of reasonably prominent foreground speech activity. This step uses S, E_n, and L and a median filter of length 0.5 s.
3. *Sentence breaking* further cuts down the segments into shorter segments at points of high confidence NON-SPEECH, assuming those would correspond to actual sentence breaks; NON-SPEECH confidences were estimated using duration and average E.

For the IHM condition, after these steps, all segments from a single microphone are assumed to be produced by a single speaker. For SDM, only the sentence

breaking step above is performed, assuming that the recognizer is the best system for discarding NON-SPEECH. The resulting segments are further tagged with speaker labels using a hierarchical agglomerative speaker clustering technique [6]. For the MDM condition, a single best channel is first chosen based on average SNR to perform further processing similar to the SDM condition.

Conference Meeting Segmentation
IHM segmentation in this subdomain is performed in 3 steps:

1. Initial label assignment is performed using our RT-04S IHM conference meeting segmenter.
2. Rather than decoding the 2-state SPEECH/NON-SPEECH activity of each participant independently [10,11], we find the best Viterbi path through a 2^K-state vocal interaction space, where K is the number of participants in the test meeting [12]. This allows us to impose constraints on the degree of overlap in each meeting [13]. Single-Gaussian, multivariate acoustic models are trained on the test data using the initial labels from (1) above, and algorithms published in [12]. Our transition model is trained on the multichannel, manual turn segmentation available from the orthographic transcription of meetings collected at the ISL. It has the form:

$$P\left(q_{t+1} = S_j \mid q_t = S_i \right) \doteq P\left(\|S_j\| \, , \, \|S_j \cap S_i\| \mid \|S_i\| \right) \tag{1}$$

   where $\|S_i\|$ and $\|S_j\|$ are the numbers of participants in SPEECH in the interaction states S_i and S_j, respectively, and $\|S_j \cap S_i\|$ is the number of participants in SPEECH in **both** S_i and S_j. A best single-participant SPEECH/NON-SPEECH path ψ_k^* is then extracted for each participant from the best multi-participant interaction path q^*.
3. Each single-participant path ψ_k^* is independently smoothed, by eliminating short intervals of speech activity and short speech activity gaps.

This algorithm significantly outperforms the segmenter used in our RT-04S evaluation system. In Table 1 we show 8th pass WER results using our RT-04S meeting recognizer on the RT-04S eval data, together with our automatic RT-04S segmentation, our automatic RT-06S segmentation, and manual segmentation. Similarly, the table gives 1st pass WER results using our RT-06S meeting recognizer on *confDEV* and *confEVAL*, with the same three segmentation systems. As this table shows, for different passes, different recognizers, and different data sets, the word error rate using our RT-06S conference meeting segmentation is 50%-80% lower than that using our RT-04S segmenter, relative to manual segmentation.

3 System Training and Development

All speech recognition experiments described in this paper were performed with the help of the Janus Recognition Toolkit (JRTk) and the Ibis single pass decoder [14].

Table 1. ASR errors committed by the last pass of our RT-04S STT system and the first pass of our RT-06S STT system, using our RT-04S meeting segmentation, our RT-06S meeting segmentation, and manual segmentation, on the IHM condition for conference meeting data

Segmentation	RT-04S, last pass				RT-06S, first pass							
	RT-04S eval data				*confDEV*				*confEVAL*			
	del	ins	sub	WER	del	ins	sub	WER	del	ins	sub	WER
RT-04S (V43c)	19.3	2.5	13.9	35.7	17.8	3.4	18.4	39.5	22.1	10.0	20.4	52.4
RT-06S (01)	14.2	1.8	14.1	30.1	13.9	3.1	20.0	37.0	15.1	5.3	21.5	42.0
manual	11.5	2.8	14.7	28.9	10.3	2.8	21.3	34.4	9.5	5.0	23.0	37.6

The following acoustic model training data was used: CMU (11hrs), ICSI (72hrs), NIST (13hrs) and AMI (16hrs) which are recordings of meetings, TED (13hrs), and CHIL (10hrs) which are recordings of lectures, and Hub4-BN (180hrs) which contains recordings of news broadcasts . All the acoustic data is in 16 kHz, 16 bit quality and recorded with head-mounted microphones, except for the CMU and Hub4-BN training data, which were recorded with either lapel or other microphones. For ICSI, NIST and AMI, farfield channels were also available.

3.1 Signal Processing

In contrast to our RT-04S system, we used two different front-ends to increase performance via cross-adaptation. The first front-end uses a 42-dimensional feature space based on MFCC with linear discriminant analysis (LDA) and a global semi-tied covariance (STC) transform [15] with utterance-based cepstral mean subtraction (CMS). It is identical to the one used in RT-04S. The second front-end replaces the Fourier transformation by a warped minimum variance distortionless response (MVDR) spectral envelope of model order 30. Due to the properties of the warped MVDR, neither the mel-filterbank nor any other filterbank was used. The advantages of the MVDR approach are an increase in resolution in low frequency regions relative to the traditionally used mel-filterbanks, and the dissimilar modeling of spectral peaks and valleys to improve noise robustness as noise is present mainly in low energy regions. Furthermore, the number of cepstral coefficients has been increased from 13 to 20. As before, a 42-dimensional feature space after LDA and a global STC transform with utterance based CMS was used.

3.2 Acoustic Model Training

The training setup was based on experiments performed during the development of the lecture translation system [1]. We selected the training data that performs best on close talking audio, using only the ICSI, NIST and TED data and skipping the CMU and the Hub4-BN training material. This reduced the WER from 36.0% to 34.8% on *lectDEV*. We also changed the model set used in RT-04S slightly by adding noise models for laughter and other human noises to

Table 2. IHM improvements over the system developed in [1] on *lectDEV*. First pass with incremental VTLN and feature-space constrained MLLR (FSA) estimation and a frame shift of 10 ms, second pass with static VTLN, FSA and MLLR and 8 ms frame shift.

Pass	Acoustic Model Training Data	#codebooks	WER
1	ICSI+NIST+CMU+TED+BN	6000	32.6
	ICSI+NIST+TED	4000	31.5
2	ICSI+NIST+CMU+TED+BN	6000	28.4
	ICSI+NIST+TED	4000	27.0

the existing breath and general noise models, and splitting the filler model into one for monosyllabic and another for disyllabic fillers.

For both subdomains, lecture and conference meetings, acoustic model training was performed with fixed state alignments, which were written by a small system (2k codebooks) trained on the corpora mentioned above. Both the MVDR and the FFT system were trained in the same way, resulting in a size of 16k distributions over 4k models, with a maximum of 64 Gaussians per model. The training was similar to that used in [1], with one modification. A second pass for incremental growing of Gaussians was performed after the STC training, which leads to an additional gain of 0.3% resulting in a WER of 32.0% on the IHM data of *lectDEV*. To train the distributions for the semi-continuous system and to compensate for the occasionally erroneous fixed-state alignments, 2 iterations of Viterbi training were performed. For the ML-SAT models, three additional iterations of maximum-likelihood speaker adaptive training (ML-SAT) [16] were run, wherein feature space adaptation and MLLR parameters were estimated for all speakers in the training set.

In addition to the FFT and MVDR systems, we trained another system using the PRONLEX phoneme set. The initial versions of the training and recognition lexica were a merger of the `callhome_english_lexicon_97061` dictionary and the LIMSI SI-284 training dictionary. Frequently missing words were added manually, and all other missing words were generated automatically with the help of a grapheme-to-phoneme conversion tool [17]. For the systems based on this phoneme set, context-independent acoustic models were trained from flat models. From them, fully context-dependent models were clustered in the same way as for the other phoneme set. The training of the context-dependent models followed the same scheme as for the other phoneme set, with the difference that 24k distributions over 3k models with a maximum of 64 Gaussians per model were used and only feature space adaptation parameters were estimated during ML-SAT.

For the lecture meeting system we used maximum a posteriori (MAP) adaptation with a weight of 0.8 for the CHIL data to adapt our semi-continuous models and gained 0.6% on top of the 32.0% WER on *lectDEV*. In a post-eval experiment, we added that data to our initial training set instead of using MAP and obtained a slight gain on *lectDEV*. During ML-SAT training, these models were applied to the CHIL training data with a weight of 4.0. Comparing the

resulting system to the system used in [1], we improved our second pass result by 1.4% absolute (see Table 2, second row).

For the conference meeting system, we used exactly the same acoustic models, except for one difference: the PRONLEX system was additionally adapted using MAP with a weight of 0.8 for the AMI training data.

For the farfield channels, we adapted the models by appending two Viterbi training iterations using the farfield ICSI and NIST meeting data to the close-talking models. Using the AMI farfield data gave no further gains on *lectDEV*.

3.3 Language Model Training

All systems use 4-gram mixture language models (LMs). Three separate LMs were trained – for lectures, non-AMI conference-style meetings, and AMI conference meetings – since the speaking style and topics were qualitatively different in these subsets. The meeting transcripts, web text, and other sources used in training were subdivided so that component LMs might be weighted differently according to style (see Table 3). Mixture weights for each LM were optimized on a held out set of data: 30k for lectures, and 53k for the AMI and non-AMI conference meetings. All the LMs were built using the SRILM-toolkit [18], with modified Kneser-Ney discounting [19]. Pruning was performed after the interpolation of the LM-components, using a fixed threshold 10^{-9}.

For web text collection, we employed two different web query strategies. For the web-L and web-M-A collections, we followed the same web text collection framework as proposed in [20], where frequently spoken 3-grams and 4-grams

Table 3. Corpora and size used in training the LM components. Data that the web collection query generation was based on is given in square brackets.

For all LMs:	
non-AMI meetings (ICSI, CMU, NIST, LDC)	1095 K
AMI meetings (RT-05S Dev: AMI-draft, AMI-final)	203 K
CHIL lectures	74 K
UW web-M [non-AMI meetings]	150M
UKA web-MP [non-AMI meetings, proceedings]	613M
w/ query-based filtering	124M
For the lecture meeting LM only:	
Translingual English Database (TED)	98 K
Hub4 Broadcast News	131M
recent speech/language proceedings (2002-2005)	130M
UKA web-L [CHIL]	146M
UKA web-LP [CHIL, proceedings]	318M
w/ query-based filtering	130M
For conference meeting LMs only:	
Switchboard CTS	4M
Fisher CTS	22M
UKA web-M-A [AMI meetings]	458M
UW web-F [Fisher CTS]	525M

from the target task training data are combined to form queries. For the other collections, the frequent n-grams from different lecture or conference meeting transcripts were combined with topic bigrams to form queries: web-MP with frequent n-grams from the conference meeting transcripts and web-LP with frequent n-grams from the lecture transcripts, respectively.[1] The goal was to obtain text reflecting a broad variety of topics, some of which are not represented in the training set. All UKA web data was perplexity filtered to 60% of the original collection sizes, with the exception of the query-based filtering where size was chosen to roughly match the UW meeting-based web collection (UW web-M).

The topic phrase generation consisted of: computing bigram tf-idf (term frequency – inverse document frequency) weights for each document in the proceedings data, zeroing all but the top 10%, averaging these weight vectors over the collection, and taking the top 1,400 bigrams excluding any with stop-words or numbers (e.g. "Section 1"). The topic bigrams were mixed randomly with the general phrases until the desired number of queries (14k) was generated.

Table 4. Perplexity (PPL) and word error rate (WER) on *lectDEV* using language models with different data source mixture components

LM	Components	PPL	WER
0	No web data	142	31.1
A	+ UW web-M	131	30.2
B	+ UKA web-L	132	30.2
C	+ UW web-M + UKA web-L	130	30.0
D	+ all web (query filtered)	128	29.9
E	+ all web (doc filtered)	126	29.8
F	(E) – UW web-M	126	29.6
G	(E) – BN,TED	126	29.7

We ran a series of experiments with different sets of web data as shown in Table 4. Not surprisingly, the biggest impact is associated with incorporating any web data, regardless of type. Using more web data gives further improvement (LMs A-B vs. C-G). Both web-L and the UW web data alone yielded similar performance (LMs A vs. B), though the web-L queries were better matched to the lecture task. However, the two are somewhat complementary and give a small gain when combined. We compared query-based vs. document-based perplexity filtering (LMs D vs. E), since some of the queries generated by randomly combining topic words and lecture n-grams effectively mixed topics. Size differences in the collections make it difficult to compare methods, but since the difference was small and document-based filtering is more flexible, we used the latter in subsequent experiments. Examining the weights used for different components of LM-E (see Table 5), we noted that a small weight was given to the UW web-M data once the other (more topic-oriented) collections were available, and we observed a small gain in performance when it was removed (LM-F). We separately

[1] The web-LP corpus includes as a subset the web-L corpus, with redundancy between the collections removed.

Table 5. Weights learned for the different component LMs for the lecture task associated with LM-E in Table 4

Speech Transcripts		Web Text		Other	
CHIL lectures	0.25	UKA web-LP	0.20	proceedings	0.19
non-AMI meetings	0.14	UKA web-MP	0.10	TED	0.004
AMI meetings	0.08	UW web-M	0.05	BN	0.004

Table 6. Weights learned for the different component LMs for the conference meeting task, with separate tuning for the non-AMI and AMI meetings

Component	LM weight	
	non-AMI	AMI
non-AMI speech	**0.31**	0.08
AMI speech	0.01	**0.42**
CHIL speech	0.002	0.005
Switchboard CTS	0.03	0.03
Fisher CTS	**0.30**	0.12
UW web-M	0.11	0.03
UW web-F	0.06	0.06
UKA web-MP	0.10	0.09
UKA web-M-A	0.07	0.16

explored removing the low weight text sources (LM-G) and again observed a small but not significant gain. Overall, the best case model reduced perplexity by roughly 10% and WER by roughly 5% relative, compared to using no web data at all. Due to time constraints, the lecture system as applied in the evaluation used LM-C for most conditions, though LM-D was used in later passes for the IHM condition. Compared to the old 4-gram LM used in [1], we gain 1.6% absolute from using LM-C, or 1.9% absolute if we use the best model obtained with subsequent development.

We did no further development for the conference meeting language models other than to introduce new web data. Based on the results of prior ICSI work [21], we did not use the BN or TED data but included CTS transcripts. In Table 6, the weights for the different language model components confirm the different nature of the AMI meetings. In addition to the expected differences of matched vs. mismatched collections, the AMI meetings do not leverage the Fisher data nearly as much as the non-AMI meetings. Interestingly, the combined weights of the different meeting-related web corpora are the same (.28) for both LMs. The overall perplexity on the two data sets is quite different (70 vs. 98 for AMI vs. non-AMI subsets of *confDEV*), though both have a WER of 31.1%.

3.4 Recognition Lexicon

For the lecture system, the dictionary contained 58.7k pronunciation variants over a vocabulary of 51.7k. The vocabulary was derived by using the corpora: BN, Switchboard, meetings (ICSI, CMU, NIST, AMI), TED and CHIL. After

applying individual word-frequency thresholds to the corpora, we filtered the resulting list with `ispell` to remove spelling errors and added a few manually checked topic words from the set of topic bigrams used in web data collection. The OOV-rate on *lectDEV* was 0.65%. The conference meeting system used a dictionary of 56k pronunciation variants over a vocabulary of 48k entries from Switchboard, Fisher, meetings, and CHIL corpora. In this case, we used the SRI vocabulary selection technique [22] available in the SRILM toolkit, again followed by `ispell` filtering and the inclusion of topic words as well as skipping those vocabulary entries not available in the lecture system dictionary.

Pronunciations for new words for most systems were generated using Festival [23]. For the PRONLEX system, pronunciations were generated automatically using Fisher's grapheme-to-phoneme conversion tool [17].

4 Experiments and Results

4.1 Decoding Strategy

In order to find the best decoding and cross-system adaptation strategy, we performed several different experiments on *lectDEV*. The best setup in terms of word error rate and complexity for all conditions uses only VTLN-trained models (VTLN) or speaker-adapted models (ML-SAT) and no speaker-independent models, even in the first decoding pass:

1. VTLN decoding using incremental, speaker-based VTLN [24] and feature-space constrained MLLR (FSA) [25] adaptation.
2. VTLN, FSA and MLLR [26] adaptation on the confidence-weighted hypothesis of the first pass and VTLN decoding with fixed adaptation parameters.
3. VTLN, FSA and MLLR adaptation on the output of second pass and ML-SAT decoding.
4. Same as in the third pass.

Using an 8 ms instead of a 10 ms frame-shift for passes 2–4, improves the final WER by about 1% absolute [9] on *lectDEV*.

In another set of experiments, we followed results presented in [27,28] and our own experience obtained during the development of a system for transcribing English European Parliament Plenary Sessions [7]. It was seen that we gain significantly (approx. 1.5% absolute) from cross-adaptation between systems with different front-ends (MVDR, FFT), and that, when cross-adaptation between MVDR and FFT leads to no further gains, cross-adapting with the PRONLEX system improves the WER after confusion network combination (CNC) [29] by 0.7% absolute [5].

4.2 Channel Combination and Selection for MDM

In RT-04S, channel combination was performed by decoding all channels and doing a confusion network combination on the resulting lattices over all channels. No selection was used, leading to a relatively high computational load for one

pass. This year, we were able to reduce the computational load by 70% with no increase in WER by performing both channel combination and selection. We constructed a single channel at the waveform level by selecting only those channels for an utterance with a high signal-to-noise ratio (SNR); this leads to an improvement in SNR of 2 dB on *lectDEV*. In addition to the speed-up on the MDM condition, we gained 4% in WER with this blind channel combination (BCC) approach compared to the SDM condition (see first and second pass overall results in Table 7). Including additional utterances and/or channels based on their SNR ratio to the confusion network combination of the BCC channel yields a further gain of 0.5% absolute. A detailed explanation is given in [30].

4.3 Overall System Performance

Table 7 lists the overall system results with automatic segmentation for RT-06S. The WERs per pass are after CNC of the lattices of the MVDR, FFT, and/or PRONLEX system used in that pass. In each pass of the IHM system, both an MVDR and an FFT system were used and cross-adapted on the previous pass. In the fourth pass, we only used the PRONLEX system and adapted the fifth pass systems (FFT, PRONLEX) on the CNC result of lattices from the third and fourth pass.

Table 7. Overall results and real-time factors on RT-05S Eval and RT-06S Eval. In contrast to previous sections, results for the conference meeting part of RT-05S Eval include meeting NIST_20050412-1303. SDM and MDM results were scored with an overlap of one.

Pass	IHM				SDM			MDM		
	lect		conf		lect		conf	lect		conf
	dev	eval	dev	eval	dev	eval	eval	dev	eval	eval
1	30.3	39.6	41.7	37.6	50.9	65.9		46.9	61.2	
2	25.0	34.7	35.2	31.9	45.9	59.0	60.1	42.0	57.0	
3	23.9	33.6	33.7	30.8	43.4	55.5	58.3	38.5	53.9	53.8
4	23.2	32.7	32.6	30.2		54.7			53.4	
5	22.9	32.2	31.9	30.2						
RTx	190				110			120		

As described above (Section 4.2), the first and second pass for the MDM condition used blind channel combination. In the third pass we added additional utterances and/or channels to the confusion network combination step. As for IHM we used both an MVDR and an FFT front-end in each pass, but in contrast to IHM, the MVDR system was adapted on the CNC result and the FFT system on the MVDR result of the subsequent pass. The first and second passes were decoded with farfield acoustic models, but in the third pass we used the close-talking acoustic models.

On the lecture meeting task, it can be seen that there is a huge gap between the development and the evaluation data results. This comes from the additional data collected by sites other than UKA. While the IHM error rates for UKA

(23.9%) and IBM (27.3%) are similar to those on the development data, which were collected by UKA only, the error rates on data from AIT (35.3%), ITC (31.8%) and UPC (54.0%) are much worse. The reason for that is likely the more interactive style of the non-UKA lecture meetings, e.g. coffee breaks (UPC), and the higher proportion of non-native speakers.

Acknowledgments

This work was partly funded by the European Union (EU) under the integrated project CHIL [31] (IST-506909).

References

1. C. Fügen, M. Kolss, D. Bernreuther, M. Paulik, S. Stüker, S. Vogel, and A. Waibel, "Open Domain Speech Recognition & Translation: Lectures and Speeches," in *ICASSP*, 2006.
2. M. Wölfel and J. McDonough, "Combining Multi-Source Far Distance Speech Recognition Strategies: Beamforming, Blind Channel and Confusion Network Combination," in *INTERSPEECH*, 2005.
3. F. Metze, Q. Jin, C. Fügen, K. Laskowski, Y. Pan, and T. Schultz, "Issues in Meeting Transcription – The ISL Meeting Transcription System," in *ICSLP*, 2004.
4. M. Wölfel and J. McDonough, "Minimum Variance Distortionless Response Spectral Estimation Review and Refinements," *IEEE Signal Processing Magazine*, September 2005.
5. S. Stüker, C. Fügen, S. Burger, and M. Wölfel, "Cross-System Adaptation and Combination for Continuous Speech Recognition: The Influence of Phoneme Set and Acoustic Front-End," in *INTERSPEECH*, 2006.
6. Q. Jin and T. Schultz, "Speaker Segmentation and Clustering in Meetings," in *ICSLP*, 2004.
7. S. Stüker, C. Fügen, R. Hsiao, S. Ikbal, Q. Jin, F. Kraft, M. Paulik, and M. W. M. Raab, Y.-C. Tam, "The ISL TC-STAR Spring 2006 ASR Evaluation Systems," in *TC-Star Workshop on Speech-to-Speech Translation*, 2006.
8. J. Makhoul, "Linear Prediction: A Tutorial Review," *Proc. of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.
9. C. Fügen, M. Wölfel, J. W. McDonough, S. Ikbal, F. Kraft, K. Laskowski, M. Ostendorf, S. Stüker, and K. Kumatani, "Advances in Lecture Recognition: The ISL RT-06S Evaluation System," in *INTERSPEECH*, 2006.
10. T. Pfau, D. P. W. Ellis, and A. Stolcke, "Multispeaker Speech Activity Detection for the ICSI Meeting Recorder," in *Proc. ASRU*, 2001.
11. S. N. Wrigley, G. J. Brown, V. Wan, and S. Renals, "Speech and Crosstalk Detection in Multichannel Audio," *IEEE Trans on Speech and Audio Processing*, vol. 13, pp. 84–91, 2005.
12. K. Laskowski and T. Schultz, "Unsupervised Learning of Overlapped Speech Model Parameters for Multichannel Speech Activity Detection in Meetings," in *Proc. ICASSP*, 2006.
13. Ö. Çetin and E. Shriberg, "Speaker Overlaps and ASR Errors in Meetings: Effects Before, During, and After the Overlap," in *Proc. ICASSP*, 2006.

14. H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A One Pass-Decoder Based on Polymorphic Linguistic Context Assignment," in *ASRU*, 2001.
15. M. J. F. Gales, "Semi-tied covariance matrices," in *ICASSP*, 1998.
16. J. McDonough, T. Schaaf, and A. Waibel, "On Maximum Mutual Information Speaker-Adapted Training," in *ICASSP*, 2002.
17. W. M. Fisher, "A Statistical Text-to-Phone Function Using Ngrams and Rules," in *ICASSP*, 1999.
18. A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *ICSLP*, 2002.
19. S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," Computer Science Group, Harvard University, Tech. Rep. TR-10-98, 1998.
20. I. Bulyko, M. Ostendorf, and A. Stolcke, "Getting more Mileage from Web Text Sources for Conversational Speech Language Modeling using Class-Dependent Mixtures," in *Proc. HLT-NAACL*, 2003.
21. Ö. Çetin and A. Stolcke, "Language Modeling in the ICSI-SRI Spring 2005 Meeting Speech Recognition Evaluation System," International Computer Science Institute, Berkeley, CA, USA, Tech. Rep. TR-05-006, 2005.
22. A. Venkataraman and W. Wang, "Techniques for Effective Vocabulary Selection," in *Proc. Eurospeech*, 2003.
23. A. W. Black and P. A. Taylor, "The Festival Speech Synthesis System: System documentation," Human Communciation Research Centre, University of Edinburgh, Edinburgh, Scotland, United Kongdom, Tech. Rep. HCRC/TR-83, 1997.
24. P. Zhan and M. Westphal, "Speaker Normalization Based on Frequency Warping," in *ICASSP*, 1997.
25. M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition," Cambridge University, Cambridge, United Kingdom, Tech. Rep., 1997.
26. C. J. Leggetter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.
27. H. Yu, Y.-C. Tam, T. Schaaf, S. Stüker, Q. Jin, M. Noamany, and T. Schultz, "The ISL RT04 Mandarin Broadcast News Evaluation System," in *EARS Rich Transcription Workshop*, 2004.
28. L. Lamel and J.-L. Gauvain, "Alternate Phone Models for Conversational Speech," in *ICASSP*, 2005.
29. L. Mangu, E. Brill, and A. Stolcke, "Finding Consensus among Words: Lattice-based Word Error Minimization," in *EUROSPEECH*, 1999.
30. M. Wölfel, C. Fügen, S. Ikbal, and J. W. McDonough, "Multi-Source Far-Distance Microphone Selection and Combination for Automatic Transcription of Lectures," in *INTERSPEECH*, 2006.
31. "CHIL – Computers in the Human Interaction Loop," http://chil.server.de.

The AMI Meeting Transcription System: Progress and Performance

Thomas Hain[1], Lukas Burget[2], John Dines[3], Giulia Garau[4], Martin Karafiat[2], Mike Lincoln[4], Jithendra Vepa[3], and Vincent Wan[1]

[1] Department of Computer Science,
University of Sheffield, Sheffield S1 4DP, UK
[2] Faculty of Information Engineering,
Brno University of Technology,Brno, 612 66, Czech Republic,
[3] IDIAP Research Institute, CH-1920 Martigny, Switzerland
[4] Centre for Speech Technology Research,
University of Edinburgh, Edinburgh EH8 9LW, UK
th@dcs.shef.ac.uk

Abstract. We present the AMI 2006 system for the transcription of speech in meetings. The system was jointly developed by multiple sites on the basis of the 2005 system for participation in the NIST RT'05 evaluations. The paper describes major developments such as improvements in automatic segmentation, cross-domain model adaptation, inclusion of MLP based features, improvements in decoding, language modelling and vocal tract length normalisation, the use of a new decoder, and a new system architecture. This is followed by a comprehensive description of the final system and its performance in the NIST RT'06s evaluations. In comparison to the previous year word error rate results on the individual headset microphone task were reduced by 20% relative.

1 Introduction

Conference room meetings are an integral basis of business life. For many they constitute a main part of their daily work. Nevertheless meetings are often viewed as ineffective, hence many attempts are made to increase effectiveness while ensuring good communication. Recordings of meetings themselves are likely to be of little help. Instead the analysis of meeting content can be used for design tools that preserve the essential information in accessible form. The foundation for such analysis is in many cases the spoken word, hence work on meeting transcription is essential for the AMI project [1]. The transcription system presented in this paper is developed by multiple sites involved in the AMI project [1].

High degrees of variability present in the meetings recordings make it an interesting task for automatic speech recognition[2]. The speaking style is conversational by nature but the presence of multiple conversation partners results in characteristic speaking style. It was found that surprisingly Broadcast News (BN) material fits reasonably well (e.g. [3]). The diversity of topics appears to

[1] http://www.amiproject.org

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 419–431, 2006.

be large, however analysis of existing corpora is ambiguous[2]. Another obvious source of variability is the recording conditions. The AMI system has focused on two conditions: the individual headset microphone (IHM) and the multiple distant microphone (MDM) conditions. While the latter seems to represent a natural situation, the former allows the establishment of baselines and assessment of the loss due to different recording setups.

In 2005 we presented our first system for participation in the NIST RT 2005 evaluations (Sys05)[4]. This initial system achieved state-of-the-art competitive performance both on conference and lecture room tasks and the system formed the basis of our development this year. Analysis of the system exhibited several issues. For example the difference in word error rate (WER) performance between manual and automatic segmentation was more than 20% relative on IHM while the difference between IHM and MDM results was approximately 30% relative. The latter was dependent on the recording setup with generally larger differences where the setup was less strictly specified. Other less prominent issues were addressed in this paper, such as speed, stability of vocal tract length normalisation (VTLN), pronunciations, adaptation of CTS models, etc.

In the following section we briefly outline the main characteristics of the 2005 system, followed by a section discussing experiments and algorithmic differences for various components in the 2006 system. This is followed by a section describing the final system architecture and results on conference and lecture room tasks. The final section concludes the paper.

2 The AMI 2005 STT System

The AMI 2005 STT system operates in a total of six passes[4][2]. The system is identical in structure both for IHM and MDM input. The systems differ in the front-ends and the acoustic models. Hence we focus initially on the description of the IHM system and highlight the differences for MDM later on.

The IHM front-end converts the recordings into feature streams, with vectors comprised of 12 MF-PLP features and raw log energy and first and second order derivatives are added. The audio stream is split into meaningful segments. The segmenter uses echo cancellation prior to classification with a multi-layer perception (MLP). After segmentation cepstral mean and variance normalisation (CMN/CVN) is performed on a per channel basis (see Fig.1).

The first decoding pass yields initial transcripts that are subsequently used for estimation of VTLN warp factors. The feature vectors and CMN and CVN are recomputed. The second pass processes the new features and its output is used to adapt models with maximum likelihood linear regression (MLLR). In the third pass word lattices are produced which are rescored with trigram language models (LMs) and meeting room specific 4-gram LMs in the fourth pass. In the fifth pass acoustic rescoring with pronunciation probabilities is performed and the lattices are compressed into confusion networks (CNs) in the final pass.

[2] Appropriate references to well known techniques mentioned in this section can be found in this paper.

Table 1. Final results with the 2005 system on the *rt05seval* test set

	TOT	Sub	Del	Ins	Fem	Male	AMI	ISL	ICSI	NIST	VT
IHM	30.6	14.7	12.5	3.4	30.6	25.9	30.9	24.6	30.7	37.9	28.9
MDM	42.0	25.5	13.0	3.5	42.0	42.0	35.1	37.1	38.4	41.5	51.1

Acoustic models are trained on the *ihmtrain05* training set which merges four meeting corpora (the NIST, ISL, ICSI corpora and a preliminary part of the AMI corpus). Model training included discriminative training and a smoothed version of heteroscedastic linear discriminant analysis (HLDA). Bigram, trigram and 4-gram language models are trained on a large variety of texts and specifically collected data by harvesting the word wide web.

The difference between MDM and IHM lies in the front-end and the acoustic model training set. The front-end operates in four stages: initial gain calibration is followed by noise compensation and frame based delay estimation between channels. The delay estimates are then used in superdirective beam-forming to yield a single output channel. All further steps were similar to the IHM case, segmentation and speaker clustering information for the MDM system were kindly provided by SRI/ICSI[3]. We repeat the results on the NIST RT 2005 conference room evaluation set (*rt05seval*) for convenience in Table 1. The system operated in 200-300 times real-time.

3 New Developments in the 2006 System

In the 2005 system we could identify a series of major and minor weaknesses of the system of which some were addressed. Further, as the 2005 system was our initial move not all components had been developed as far as we would have liked and hence we also continued on this path to include new technologies. The main sets of changes to the system include: Improved segmentation for IHM; standard unsmoothed HLDA with removal of silence; posterior probability based features [5]; speaker adaptive training (SAT) with constrained MLLR (CMLLR) [6]; acoustic feature space mappings and maximum-a-posteriori (MAP) adapted HLDA; search model based LM data collection; as well as a modified system architecture that includes the use of a new decoder. In the following sections we present more details on these changes.

3.1 Improved Front-Ends

Several changes to both the IHM and MDM front-ends (see Figure 1) were made.

Individual Headset Microphone. The initial cross-talk suppression is based on an adaptive LMS echo canceller [7] followed by MF-PLP feature extraction. Different to last year, features to aid in the detection of cross-talk are extracted from the original recording (prior to cross-talk suppression). These features are cross-channel normalised energy, signal kurtosis, mean cross-correlation and maximum

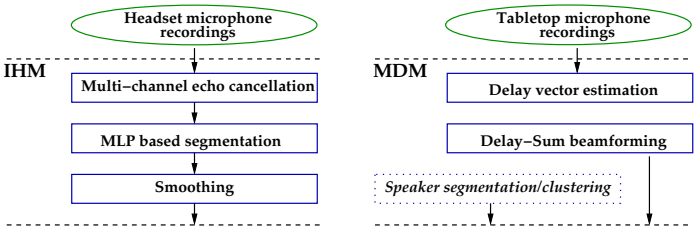


Fig. 1. Front-ends for both IHM and MDM conditions

Table 2. WER on *rt06seval IHM* first passes using manual and automatic segmentation

Segmentation	TOT	EDI	TNO	CMU	VIT	NIS
manual	40.4	32.8	45.4	43.4	41.6	40.6
automatic	41.4	33.7	45.9	43.4	43.6	42.5

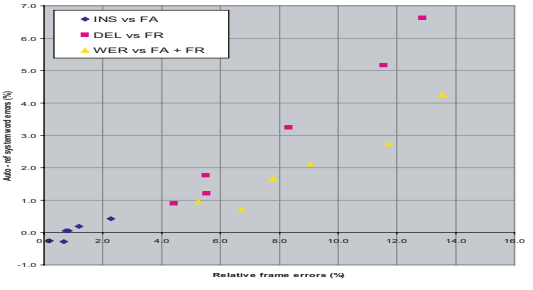
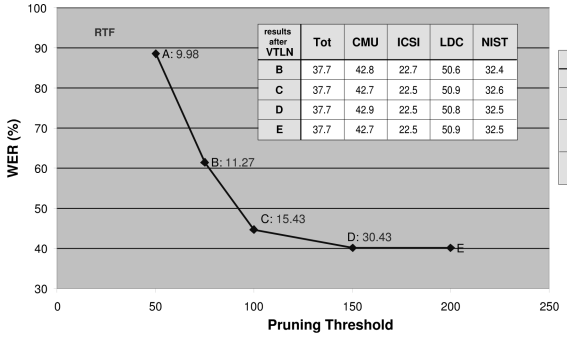


Fig. 2. False Alarm (FA) and False Reject (FR) frame error rate changes in relationship to WER changes between manual and automatic segmentation on *rt06seval*

normalised cross-correlation. The cross-channel normalised energy is calculated as the energy for the present channel divided by the sum of energies across all channels [8]. In addition the MLP setup was changed to include 50 hidden units and the models are trained on 90 hours of data from all meetings in the *ihmtrain05* set. On *rt05seval* the first pass was found to give almost identical results to manual segmentation. Table 2 shows a comparison of manual versus automatic segmentation on *rt06seval*.

Figure 2 shows the correlation between WER and frame error rates for the meetings in *rt06seval*. The sum of false alarm (FA) and false reject (FR) rates exhibit a linear relationship with word errors. The main contributor are FR errors, which are unrecoverable.

Multiple Distant Microphones. Only minor changes were made. Analysis on *rt05seval* showed that the system performed poorly on recordings from the VT meeting room. The reason was the use of only two microphones that were placed



results after VTLN	Tot	CMU	ICSI	LDC	NIST
B	37.7	42.8	22.7	50.6	32.4
C	37.7	42.7	22.5	50.9	32.6
D	37.7	42.9	22.5	50.8	32.5
E	37.7	42.7	22.5	50.9	32.5

Warping factorsRMSE

	B	C	D	E
B	-	-	-	-
C	0.0164	-	-	-
D	0.0162	0.0023	-	-
E	0.0164	0.0023	0.0001	-

Fig. 3. WER results on *rt04seval IHM* in the second pass. Real time factors (RTFs) combine first and second pass. The table shows RMSE results for operating points in the first and second pass. RMSE denotes the root mean squared warp factor difference to the baseline system.

Table 3. WER results for SAT rescoring 4-gram lattices on *rt05seval* IHM. LCRC denotes posterior based features.

SAT iterations	PLP	PLP+LCRC
-	28.7	25.2
adapt	27.9	24.2
1	27.6	24.1
2	27.4	24.0

far apart in the room, causing delay estimation and hence poor beam-forming. The solution was to simply pick the channel with the highest energy for every time frame. This approach was also beneficial for the *rt06seval* set where the VT recording setup included four microphones directed at the speakers. Further problems had been caused by mis-aligned audio files, a problem eliminated in our 2006 system. Overall these changes brought improvements of 2.2% WER absolute on *rt05seval* in the first pass, and a 6% absolute change on VT data.

3.2 Vocal Tract Length Normalisation Experiments

Maximum likelihood based VTLN was part of the 2005 system, where relative WER improvements of more than 10% for both IHM and MDM were found. However, the cost in terms of complexity and real time was large, since the first pass is only devoted to finding initial transcripts for VTLN. Experiments were conducted to determine the importance of high quality transcripts. Fig. 3 shows WER results in the second pass as a function of real time factors and associated pruning in the first pass. On the right side the effect of pruning in both passes on the warp factor estimates compared to the baseline is shown. The operating point C was chosen for the first pass and D for the second pass.

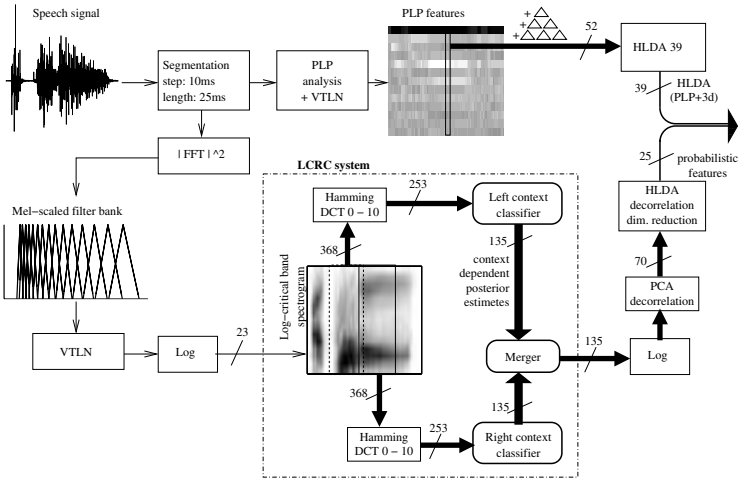


Fig. 4. Computation of LCRC features

Table 4. WER results on *rt05seval*/IHM rescoring Sys05 4-gram lattices. Contrasting LCRC features using SAT and MPE.

System	Training critrion	PLP	LCRC+PLP
Baseline	ML	28.7	25.2
SAT	ML	27.6	23.9
SAT	MPE	24.5	21.7

3.3 Speaker Adaptive Training

The system already makes use of multiple speaker and channel normalisation techniques. Both CMN/CVN and VTLN yield substantial gains in WER close to 20% relative. Both techniques are simple and have few parameters. Speaker adaptive training (SAT) allows further normalisation and was already successfully applied in [3]. Here constrained MLLR[6] with two transforms was used where one is designated to the silence models.

3.4 Posterior Based Features

MLP based features have been deployed in large ASR systems (e.g. [9]). In [5] a similar approach was taken to produce a set of posterior probability based features computed with multiple layers of MLPs. While these kinds of features do not yield good performance directly, they clearly hold complementary information to the standard PLPs or MFCCs, and can bring substantial improvements in WER. Figure 4 describes the creation process of the feature vector. The top part shows standard MF-PLP generation and projection into a 39 dimensional space using HLDA. For the generation of the LCRC features first standard VTLN and CMN/CVN is applied to Mel frequency log filterbank (FB) coefficients. 23

FB coefficients are extracted every 10ms and 15 vectors of left context are then used to find the LC state level phone posterior estimates. The same procedure is performed with the right context. These posteriors are then combined with a third MLP network and after logarithmic compression the 135 dimensional feature vector is reduced to dimension 70 using principal component analysis. This step is only necessary because the final dimensionality reduction using HLDA was not feasible with such high dimensional vectors. The final 25-dimensional feature vector is appended to the standard 39 dimensional feature. Mean and variance normalisation is repeated at this stage.

Table 4 shows results for different combinations of training strategies and feature vectors. The number of states and mixture components remained constant and all systems use VTLN. Despite a considerable increase in dimensionality and data sparsity substantial performance improvements was found. The gains appear to be independent of the underlying training strategies. Note that the results in Table 4 may be somewhat biased because they were obtained by lattice rescoring.

3.5 Adapting to the Meeting Domain

One of the main short-comings of the 2005 system was the fact that only meeting data (*ihmtrain05/mdmtrin05*) could be used for discriminative training. Both sets are comparatively smaller than the CTS training set and experimental evidence suggests that discriminative training should perform better ([10]) with more data. However, CTS and meeting data have different bandwidth and initial experiments showed that joint adaptation and projection into common space yields better performance[1]. The use of HLDA complicates matters considerably since it is not clear in which domain the matrix should be trained. It was decided to project the meeting data into the narrowband space where both HLDA statistics can be gathered and discriminative training be performed without regeneration of training lattices.

Initial full covariance statistic is estimated on the CTS training set. A single CMLLR transform is trained to map the 52D wideband (WB) meeting data to a 52D narrowband (NB) CTS space. The meeting data is mapped with this transform and full covariance statistics is obtained using models based on CTS phonetic decision tree clustering. The two sets of statistics are combined with MAP-like equations. The combined set of statistics is used to obtain a joint HLDA transform (JT). Now combined models in JT space can be trained using both CTS and mapped meeting data. These are then used to retrain CTS models in JT space, followed by speaker adaptive training and minimum phone error (MPE) training[10]. Equivalently to adaptation of maximum likelihood models with MAP, the JT/SAT/MPE models are adapted to meeting data using MPE-MAP[11]. The inclusion of SAT requires the presence of transforms on meeting data. These are obtained from SAT training of MAP adapted CTS models in JT space. Overall the performance improvement of this procedure was at least 0.6% on rt06seval. However, the elaborate process prohibited inclusion of LCRC features at this point.

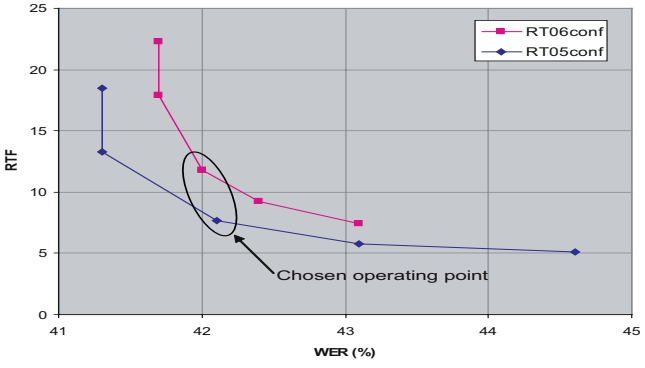


Fig. 5. WER results on IHM data with the first pass using Juicer

3.6 Web Data Collection

Language model data for meetings are a rare resource and hence all sites have included language model material collected from the world wide web using approaches as originally described by [12]. The technique is based on sending a set of queries to well known Internet search engines and harvesting of the resulting documents. In Sys05 we used data collected using only queries (n-grams) that were not already present in our background LM material. Since then we have refined this approach [13]. In this work we use search models to predict the benefit of the search results on perplexity. The set of N-grams (word w with history h) present in a sample text T is ranked inversely with

$$\sum_{v} \frac{(\alpha P(w|h,T) + \beta P(w|h,B))}{(\alpha P(v|h,T) + \beta P(v|h,B))}$$

The probability estimates $P(w|h,T)$and $P(w|h,B)$ are provided by language models trained on the sample text and the background material B. The weights α and β were set equal. Small gains in perplexity were found with moderate data set sizes.

3.7 Juicer

As already discussed above, the 2005 system had a high RTF, partly due to slow initial stages. A second approach to address this is a faster decoder. Juicer [14] is a large vocabulary speech decoder based on weighted finite-state transducer (WFST). It uses a time-synchronous Viterbi search based on the token-passing algorithm with beam-search and histogram pruning. Juicer works with a single WFST composed of language model, dictionary and acoustic model. For the composition and optimisation of WFST resources, Juicer relies on the functionality of the AT&T finite-state machine library [15] and MIT FST toolkit [16]. The main advantage of WFST-based decoders is the decoupling of the decoding network generation and the actual decoding process. But there are limitations

in composing the decoding networks, mainly due to high memory requirements, when used with large higher-order N-gram language models. Hence, pruned trigram language models were used for constructing decoding networks. Figure 5 shows performance versus RTF. The overall performance is within 1% absolute of the best results with HDecode[3].

4 System Architecture

Figure 6 shows the 2006 system architecture in diagrammatic form. In comparison to Sys05 the following major changes were made: The initial pass P1 now includes posterior feature computation; the output of P1 is used for both VTLN and adaptation of SAT models in pass P2. Lattice generation is performed in P2, with lattice expansion to unified 4-gram lattices in P3. These lattices are then rescored with different acoustic models. The original plan was to perform system combination by combining confusion networks, however this turned out to yield poorer performance. The best performing path was P4b followed by P5a. These passes are similar to the lattice rescoring passes in sys05, however include standard MLLR adaptation on top of the use of constrained MLLR.

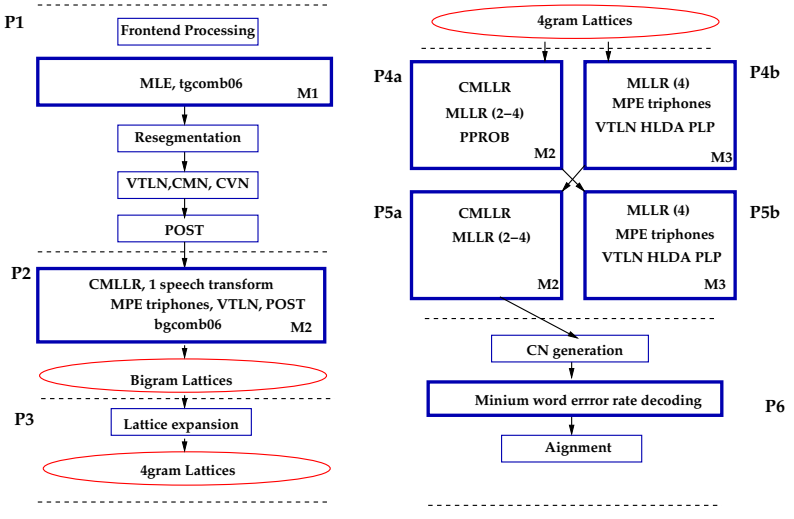


Fig. 6. Processing steps of the AMI 2006 system

5 System Components

Most of the system software is based on HTK. In particular three different decoders were included: In P1 Juicer (see Section 3.7) is used; in P2 HDecode provides lattice generation capability; passes P4 and P5 operate with HVite for rescoring. Confusion networks are generated using the SRI LM toolkit.

[3] HDecode is distributed as part of HTK (`http://htk.eng.cam.ac.uk`)

Table 5. Language model perplexity results on *rt05seval*

Perplexity	conference			lecture		
	2g	3g	4g	2g	3g	4g
2006 LM	106.9	86.2	82.7	157.9	127.6	122.4
2005 LM	105.6	84.3	81.2	165.6	137.4	134.5

Table 6. WER results of Sys06 on *rt05seval*

	TOT	Sub	Del	Ins	Fem	Male	AMI	CMU	ICSI	NIST	VT
IHM	23.7	12.0	9.9	1.7	23.7	20.3	22.0	20.1	21.1	30.0	25.7
MDM	33.0	18.7	12.3	2.1	33.0	35.4	28.8	32.6	35.8	35.4	33.7

5.1 Acoustic Models

All acoustic models are phonetically state tied mixture of Gaussian HMMs with 16 mixture components. For the IHM system models were trained on *ihmtrain05*, for MDM models were trained on *mdmtrain05*. LCRC MLP models are trained on 30 hour subsets. The models M1 (see Figure 6) are identical to those used in Sys05. Models M2 are trained on PLP+LCRC features using SAT and MPE as outlined above. Several iterations of SAT were necessary for improved performance, followed by a total of 15 iterations of MPE training. For IHM models both MMI and MPE numerator and denominator statistics were combined with fixed weights. The M3 models are trained in the form outlined in Section 3.5, on the 300 hour *h5train03* training set[4].

5.2 Vocabulary, Language Models and Dictionaries

The vocabulary was built in similar fashion to Sys05 and changed only moderately. New web-data was collected for both conference and lecture room meetings with the technique outlined in Section 3.6. Table 5 shows perplexity results on *rt05seval* (both conference and lecture room meetings). Note that the 2005 language models shows lower perplexity. The reason for this behaviour is that the 2006 LM interpolation weight estimation did not include ICSI data as it was not part of *rt06seval*. The new method of data collection appeared to work well on lecture room data.

6 Overall System Performance

The development performance of the AMI 2006 system (Sys06) on the *rt05seval* data set is shown in Table 6 and can be directly compared with the results shown in Table 1. It is clear that at least on this test set substantial improvements have been made. The main improvements of the IHM system appear on the AMI data, while MDM improvement is highest on the VT subset.

Table 7 shows IHM results on the 2006 evaluation set, both with automatic and manual segmentation. The huge difference between initial and final pass

Table 7. WER results with Sys06 on rt06seval

	Automatic Segmentation						Manual segmentation					
	TOT	CMU	EDI	NIST	TNO	VT	TOT	CMU	EDI	NIST	TNO	VT
P1	42.0	41.9	41.0	39.0	42.1	44.8	40.3	40.4	39.5	38.7	37.6	40.9
P2a	29.2	29.2	27.4	27.7	29.5	32.4	26.5	26.7	25.5	26.6	22.3	28.8
P3.tg	26.6	26.3	25.2	25.7	27.0	29.9	21.1	21.2	19.7	21.8	17.0	23.9
P3	26.0	25.7	24.6	25.2	26.3	29.5	22.9	22.9	22.3	23.8	19.0	25.1
P4a	25.1	25.0	22.8	23.8	26.0	29.1	21.9	21.9	20.7	22.6	18.1	24.6
P4b	25.6	25.3	23.8	24.9	24.3	29.8	22.5	22.5	21.8	23.6	17.2	25.6
P5a	24.6	24.4	22.6	23.6	24.1	28.8	21.5	21.5	20.3	22.4	17.1	24.2
P5a-cn	24.2	24.0	22.2	23.2	23.6	28.2	21.1	21.2	19.7	21.8	17.0	23.9

Table 8. WER results for Sys06/MDM on the RT06 conference and lecture room test sets

	Conference				Lecture			
	TOT	Sub	Del	Ins	TOT	Sub	Del	Ins
P1	58.2	35.8	16.7	5.7	70.7	46.0	16.3	8.5
P2a	45.6	26.4	15.1	4.1	60.0	31.6	23.6	4.9
P3	42.0	24.5	13.2	4.4	58.2	30.8	22.0	5.4
P4a	41.7	22.9	14.9	3.9	57.8	28.5	24.3	4.9
P5	40.9	22.2	15.3	3.5	56.1	28.2	23.9	4.0

results is even larger than before due to faster processing. After the third pass the results are already very close to the final performance, especially for manual segmentation. Even though the P4b system has lower performance on its own the inclusion into the adaptation path yields a further 0.5% absolute. Simple adaptation with P4a supervision did not give any improvement. It is interesting to note that automatic and manual segmentation differ minimally initially however the difference is immediately obvious once the systems use adaptation. Since this cannot be a speaker labelling problem it is likely that this is caused by cutting into sentences.

The MDM performance is given in Table 8 (non-overlap results). Again the initial pass yields very poor performance and the difference between the output of the third pass and the final result is small. Overall the gap between IHM and MDM performance appears to be wider than on the *rt05seval* test set.

6.1 Lecture Room Meetings

Similar to Sys05, the only component changed to the conference room meeting transcription system was the language model. Neither dictionary nor any acoustic models were modified. Results for both IHM and MDM can be found in Tables 9 and 8 respectively. Note the poor performance of the initial pass of both systems. In the IHM case however the system recovers reasonably well. Substantial difference in WER between data sources is visible. Further the high

Table 9. WER results for Sys06/IHM on lecture room data

	TOT	Sub	Del	Ins	AIT	IBM	ITC	UKA	UPC
P1	88.2	10.4	68.2	9.6	100.5	60.4	89.6	92.7	92.3
P2a	39.2	21.8	7.3	10.2	66.3	34.8	37.4	29.9	44.5
P3	37.1	20.1	6.6	10.4	63.6	31.6	35.6	28.6	41.6
P4a	35.1	19.2	6.9	9.0	56.5	31.5	33.5	27.5	39.9
P4b	37.2	20.5	6.7	10.1	66.1	32.0	36.5	28.5	39.8
P5a-cn	33.4	18.6	6.3	8.4	56.6	29.1	32.1	25.6	37.6

deletion rate for MDM is unusual and is not mirrored in the conference room data.

7 Conclusions

We have presented the changes made to the AMI 2005 system for transcription of speech in meetings. The main performance improvements originate for improved front-ends both in terms of segmentation and feature generation. As in 2005, there is still a large gap between performance on automatic and manual segmentation. This and the large difference between IHM and MDM results will require increased attention. We have also made improvements towards a faster system with real time factors below 100.

Acknowledgements

This work was largely supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811). We also would like to thank Andreas Stolcke and ICSI for providing the segments and speaker labels for MDM data.

References

1. Hain, T., Burget, L., Dines, J., Garau, G., Karafiat, M., Lincoln, M., McCowan, I., Moore, D., Wan, V., Ordelman, R., Renals, S.: The development of the AMI system for the transcription of speech in meetings. In: Proc. MLMI'05. (2005)
2. Hain, T., John Dines and, G.G., Karafiat, M., Moore, D., Wan, V., Ordelman, R., Renals, S.: Transcription of conference room meetings: an investigation. In: Proc. Interspeech'05. (2005)
3. Stolcke, A., Anguera, X., Boakye, K., Cetin, O., Grezl, F., Janin, A., Manda, A., Peskin, B., Wooters, C., Zheng, J.: Further progress in meeting recognition: The icsi-sri spring 2005 speech-to-text evaluation system. In: Proc. NIST RT'05 Workshop. (2005)
4. Hain, T., Burget, L., Dines, J., Garau, G., Karafiat, M., Lincoln, M., McCowan, I., Moore, D., Wan, V., Ordelman, R., Renals, S.: The 2005 AMI system for the transcription of speech in meetings. In: Proc. NIST RT'05 Workshop, Edinburgh (2005)

5. Schwarz, P., Matìjka, P., Cernocký, J.: Towards lower error rates in phoneme recognition. In: Proc. of 7th Intl. Conf. on Text, Speech and Dialogue. Number ISBN 3-540-23049-1 in Springer, Brno (2004) 8
6. Gales, M.J.F.: Linear transformations for hmm-based speech recognition. Technical Report CUED/F-INFENG/TR-291, Cambridge University Engineering Department (1997)
7. Messerschmitt, D., Hedberg, D., Cole, C., Haoui, A., P.Winship: Digital voice echo canceller with a tms32020. Application report SPRA129, Texas Instruments (1989)
8. Wrigley, S., Brown, G., Wan, V., Renals, S.: Speech and crosstalk detection in multichannel audio. IEEE Trans.Speech and Audio Processing **13**(1) (2005) 84–91
9. Zhu, Q., Chen, A.S.B., Morgan, N.: Using MLP features in sri's conversationl speech recognition system. In: Proc. Interspeech'05. (2005)
10. Povey, D.: Discriminative Training for Large Vocabulary Speech, Recognition. PhD thesis, Cambridge University (2004)
11. Povey, D., Gales, M.J.F., Kim, D.Y., Woodland, P.C.: MMI-MAP and MPE-MAP for acoustic model adaptation. In: Proc. Eurospeech'03. (2003)
12. Bulyko, I., Ostendorf, M., Stolcke, A.: Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures. In: Proc. Human Language Technology Conference 2003. (2003)
13. Wan, V., Hain, T.: Strategies for language model web-data collection. In: Proc. ICASSP'06. Number SLP-P17.11 (2006)
14. Moore, D., Dines, J., Doss, M.M., Vepa, J., Cheng, O., Hain, T.: Juicer: A weighted finite state transducer speech decoder. In: Proc. MLMI'06. (2006)
15. Mohri, M., Pereira, F., Riley, M.: General-purpose finite-state machine software tools. Technical report, AT&T Labs -Research (1997)
16. Hetherington, L.: The mit fst toolkit. Technical report, L. Hetherington, "The MIT FST toolkit", MIT Computer Science and Artificial Intelligence Laboratory: http://people.csail.mit.edu/ilh/fst, May 2005. (2005)

The IBM Rich Transcription Spring 2006 Speech-to-Text System for Lecture Meetings

Jing Huang, Martin Westphal, Stanley Chen, Olivier Siohan, Daniel Povey, Vit Libal, Alvaro Soneiro, Henrik Schulz, Thomas Ross, and Gerasimos Potamianos

IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598, U.S.A.

Abstract. We describe the IBM systems submitted to the NIST RT06s Speech-to-Text (STT) evaluation campaign on the CHIL lecture meeting data for three conditions: Multiple distant microphone (MDM), single distant microphone (SDM), and individual headset microphone (IHM). The system building process is similar to the IBM conversational telephone speech recognition system. However, the best models for the far-field conditions (SDM and MDM) proved to be the ones that use neither variance normalization nor vocal tract length normalization. Instead, feature-space minimum-phone error discriminative training yielded the best results. Due to the relatively small amount of CHIL-domain data, the acoustic models of our systems are built on publicly available meeting corpora, with maximum a-posteriori adaptation applied twice on CHIL data during training: First, at the initial speaker-independent model, and subsequently at the minimum phone error model. For language modeling, we utilized meeting transcripts, text from scientific conference proceedings, and spontaneous telephone conversations. On development data, chosen in our work to be the 2005 CHIL-internal STT evaluation test set, the resulting language model provided a 4% absolute gain in word error rate (WER), compared to the model used in last year's CHIL evaluation. Furthermore, the developed STT system significantly outperformed our last year's results, by reducing close-talking microphone data WER from 36.9% to 25.4% on our development set. In the NIST RT06s evaluation campaign, both MDM and SDM systems scored well, however the IHM system did poorly due to unsuccessful cross-talk removal.

1 Introduction

Integrated Project CHIL ("Computers in the Human Interaction Loop"), funded by the EU Information Society Technologies programme, aims to create people-friendly computing by monitoring how people interact, exchange information and collaborate to solve problems, learn, or socialize in meetings. These goals are not possible without a detailed understanding of the human state, human activities, and intentions. An important initial step towards this goal is to be able to automatically generate transcripts of the conversational speech under "always-on" audio capturing from far-field (non-intrusive) microphone sensors.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 432–443, 2006.

Conversational speech recorded by distant microphones in noisy reverberant settings poses significant challenges to state-of-the-art automatic speech recognition (ASR) technology. Furthermore, a number of factors such as multiple speakers with often overlapping speech, as well as the non-native accents and technical content of CHIL seminars create additional ASR challenges. With limited in-domain CHIL data available, it is natural to leverage useful out-of-domain data such as the Fisher and broadcast news corpora, available through the Linguistic Data Consortium (LDC) [1]. To face these challenges, during the first CHIL-internal ASR evaluation campaign in January 2005 ("CHIL evaluation run #1"), we opted to employ a number of in-house available acoustic models and combine them using the ROVER technique [2], after adaptation on limited CHIL data [3]. For language modeling, we used a small amount of meeting transcripts and conference proceeding texts, interpolated with Fisher data. However, the approach proved ineffective, resulting for example in the relatively high word error rate of 36.9% on close-talking data [3]. Furthermore, such system would not have been allowed entry into the NIST RT06s evaluation, due to its use of non-publicly available data sources (in addition to LDC corpora) for acoustic model training.

As a result, we have opted to start development "from scratch" for participating in this year's NIST-sponsored RT06s campaign. We directed most of our effort in organizing most publicly available meeting corpora [1] and in training acoustic models based exclusively on such data. The effort was augmented by training and optimizing language models appropriate for the task. In contrast, no effort has been made to develop and incorporate front-end processing techniques for signal-space noise reduction and combination of distant (far-field) microphones. Overall, the same training procedures were shared between distant and individual headset microphone conditions, of course with a number of small variations. Our progress in this effort was benchmarked over the test set of the first CHIL-internal evaluation ("CHIL evaluation run #1"), which constitutes a subset of the RT05s development data.

The remainder of the paper is structured as follows: The data resources used are described in Section 2. Section 3 describes the system basics including the front end, segmentation, acoustic models, lexicon, and the language model. Results with comments are presented in Section 4, with Section 5 concluding the paper.

2 Data Resources

We briefly describe the corpora used for training, development, and evaluation of our acoustic models for both far-field and close-talking conditions. In addition to these, a number of publicly available text sources were also used for language modeling, as discussed in Section 3.4.

2.1 Training Data

The following meeting resources, available to all RT06s participant sites, were used for acoustic model training:

- ICSI meeting data, about 70 hours.
- NIST meeting pilot corpus, about 15 hours.
- RT04 development and evaluation data, about 2.5 hours.
- RT05s development data (excluding the CHIL 2005 evaluation data – see Section 2.2), about 4.5 hours.
- AMI meetings, about 16 hours.
- CHIL 2006 development data, about 3 hours.
- Additional CHIL data from the CHIL evaluation dry run (June 2004) and an intermediate collection during the Summer of 2004, for a total of about 4 hours.

With the exception of the last source (no far-field data were used), all other datasets provided both close-talking and multiple far-field microphone data. For training on the latter, we selected all table-top microphones present in the corpora. An exception was made for the AMI data, where four microphones from the eight-element circular microphone arrays were selected based on their location in the room or their SNR estimate. This approach resulted to approximately 473.5 hours of training data for the far-field MDM/SDM systems. For the IHM condition, all available close-talking channels were used, resulting to about 124 hrs of training. Notice that additional available and relevant data to this task, such as the RT05s evaluation data set and the TED corpus [1] were not used due to lack of time.

2.2 Development and Evaluation Data

In order to benchmark improvements and guide our development, we chose as development data the evaluation set of the CHIL-internal evaluation of January 2005 ("CHIL evaluation run #1"). This set provided us with about 1.8 hours of IHM data and four table-top microphones for a total of 8.7 hours. The set was later made available as part of the development data for the NIST RT05s evaluation campaign of last year, and it will be further referred to in this paper as the "CHIL eval05" set.

For evaluation, the lecture meeting data part of the RT06s test set was used. This contained a total of 190 minutes of data recorded in the smart rooms of five CHIL sites: AIT, IBM, ITC, UKA, and UPC. Multiple headset and table-top microphones were present in these data, with the choice of microphones to be used in the evaluation designated by NIST. IBM participated in three conditions, namely the:

- *Mutiple distant microphone* (MDM) condition, which constituted the *primary* condition of this evaluation, with typically all table-top microphones allowable for use;
- *Single distant microphone* (SDM), with only one table-top microphone allowed to be used, specified by NIST; and the
- *Individual head microphone condition* (IHM), where all headset microphone channels were to be decoded.

Note that in contrast to the MDM and SDM conditions, where all speech was to be decoded, in the IHM condition the purpose was to recognize the speech of the person wearing the headset (i.e., decoding cross-talk was penalized).

3 The IBM Systems

For this evaluation, two main systems were developed by the IBM team: One for far-field acoustic conditions, with obvious MDM- and SDM-condition specific variants, and one system for the IHM condition. Both used an identical language model and a quite similar acoustic model training procedure. Details are described in the next sub-sections.

3.1 Front-End, Segmentation, and Cross-Talk Removal

The features used to represent the acoustic signal for recognition are 40 dimensional vectors obtained from a linear discriminant analysis (LDA) projection. The source space for the projection is 117-dimensional and is obtained by concatenating 9 temporally consecutive 13-dimensional acoustic observation vectors based on perceptual linear prediction (PLP). The PLP features are computed at a rate of 100 frames per second from a Hamming windowed speech segment of 25 ms duration. The vectors contain 13 cepstral parameters obtained from the LPC analysis of the cubic root of the inverse DCT of the log outputs of a 24-band, triangular filter bank. The filters of this bank are positioned at equidistant points on the Mel-frequency scale between 0 and 8 kHz. The cepstral parameters are mean-normalized on a per-speaker basis. No noise filtering was applied to either MDM data or IHM data.

For segmentation, the following procedure is employed: We first segment audio files into speech and non-speech segments, followed by deletion of the non-speech segments. We use an HMM-based segmentation system that models speech and non-speech segments with five-state, left-to-right HMMs with no skip states. The output distributions in each HMM are tied across all states in the HMM, and are modeled with a mixture of diagonal-covariance Gaussian densities. The speech and non-speech models are obtained by applying a likelihood-based, bottom-up clustering procedure to the speaker-independent acoustic model. The speech segments are then segmented into homogeneous segments using Ajmera and Wooters' change-point detection procedure [4]. This is followed by a clustering procedure to cluster the segments into pseudo-speaker clusters that can then be used for speaker adaptation. We use the following clustering mechanism: All homogeneous speech segments are modeled using a single Gaussian density, and are clustered into a pre-specified number of clusters using K-means and a Mahalanobis distance measure. For the lecture meeting data, the number of speaker clusters is set to four.

For cross-talk removal, a simple algorithm is used to pre-process the close-talking microphone signal, before passing it to a speech recognizer. The objective is to detect the signal segments containing foreground speech and, in effect, reduce the insertion error rate of the recognizer, which typically has difficulty

distinguishing foreground (further also referred as "speech") from background speech or noise (further also referred as "non-speech"). The algorithm consists of two parts:

- Frame labeling, where each frame is independently labeled as speech or non-speech. Frames are overlapping segments of signal assumed homogeneous with respect to speech/non-speech signal analysis.
- Post-processing, where frames are grouped into larger speech or silence segments.

The frame labeling algorithm is very similar to [5]. There are specific situations, however, where the algorithm fails, for example when the foreground speaker talks very quietly, or the microphone is incorrectly placed relatively far from the wearer's mouth.

3.2 Acoustic Modeling

The speaker-independent (SI) acoustic model is trained on 40-dimensional features generated by an LDA projection of PLP features (see previous section), mean normalized on a per-speaker basis. The SI model uses continuous density, left-to-right HMMs with Gaussian mixture emission distributions and uniform transition probabilities. The number of mixtures for a tied state s with C_s observations is given by $4 \times C_s^{0.2}$. The final mixture distributions are obtained as the result of a splitting procedure, starting with single Gaussian distributions. Intermediate mixture distribution estimates are obtained using the expectation-maximization (EM) algorithm updating the mixture weights, means and covariances. In addition, the model uses a global semi-tied covariance [6,7] linear transformation, which is also updated at every EM training stage. The sizes of the mixtures are increased in steps interspersed with EM updates until the final model complexity is reached. Each HMM has three states, except for the silence HMM, which is a single-state model. The system uses 45 phones, among which 41 are speech phones, one is the silence phone, and three are noise phones, modeling background noise, vocal noise, and breathing noise. The MDM HMMs use 6,000 context-dependent tied state distributions, obtained by decision tree clustering of quinphone statistics using context questions based on 73 phonetic classes. The total number of Gaussian densities is about 200k. The IHM system is somewhat smaller, with 5000 context-dependent tied states and about 120k Gaussians. Since only 5% of the training data is from CHIL, MAP-adaptation of the SI model was deemed necessary to improve performance on CHIL data.

From this SI model, three different MDM models are derived based on whether variance normalization and/or vocal tract length normalization (VTLN) [8,9] are used. The three speaker adaptive training (SAT) [6,7] models are described in the following:

- **Model A:** This model is trained directly after the SI model on features in a linearly transformed feature space resulting from applying fMLLR transforms to the SI features. fMLLR transforms are computed on a per-speaker basis for all speakers in the training set.

- **Model B:** The SI features are further normalized with a voicing model (VTLN) with no variance normalization. The frequency warping is piecewise linear using a breakpoint at 6500 Hz. The most likely frequency warping is estimated from among 21 candidate warping factors ranging from 0.8 to 1.2 using a step of 0.02. Warping likelihoods are estimated using a voicing model. This model uses the same state tying as the SI model, however its states model emissions using full covariance Gaussian distributions, and the model is based on 13-dimensional PLP features.

  A VTLN model is subsequently trained on features in the VTLN warped space. VTLN warping factors are estimated on a per-speaker basis for all data in the training set using the voicing model. In that feature space, a new LDA transform is estimated and a new VTLN model is obtained by decision tree clustering of quinphone statistics. The HMMs have 10k tied states and 320k Gaussians.

  Following VTLN, SAT Model B is trained on features in a linearly transformed feature space resulting from applying fMLLR transforms to the VTLN normalized features. fMLLR transforms are computed like the VTLN warping factors on a per-speaker basis for all speakers in the training set. The HMMs have 10k tied-states and 320k Gaussians.
- **Model C:** SAT Model C is trained with the same procedure as Model B, except that variance normalization is now applied.

Following training of SAT models A, B, and C, we estimate feature-space minimum phone error (fMPE) transforms [10] for all three. The fMPE projection uses 1024 Gaussians obtained from clustering the Gaussian components in the SAT model. Posterior probabilities are then computed for these Gaussians for each frame, and time-spliced vectors of these posterior probabilities are the foundation for the features that are subjected to the fMPE transformation. The fMPE transformation maps the high-dimensional posterior-based observation space to a 40-dimensional fMPE feature space. The MPE model is then trained in this feature space with MAP/MPE on the available amount of CHIL-only data [11].

Note that in contrast to the MDM/SDM system, the IHM system is only trained with the procedure in Model C. A total of 5600 context-dependent states and 240k Gaussians are used in this IHM system.

3.3 Recognition Process for MDM Data

After the automatic segmentation and speaker clusters are determined, for each table-top microphone, a final system output is obtained in 3 passes:

a. The SI pass uses MAP-adapted SI models to decode.
b. Using the transcript from step a., warp factors are estimated for each cluster using the voicing model, and fMLLR transforms are estimated for each cluster using the SAT model. The VTLN features after applying the fMLLR transforms are subjected to the fMPE transform, and a new transcript is obtained by decoding, using the MPE model and the fMPE features. The MPE is also trained with MAP on the CHIL data.

c. The lattices resulting from step b. are rescored using an interpolated language model. The one-best at this step will be referred to as CTM-n, where n stands for model A, B, or C.

Clearly, we have three different MPE models at step b, therefore we have three outputs CTM-A, CTM-B, CTM-C for *each* available table-top microphone.

The final output for MDM is obtained using ROVER in the following way: First, for each table-top microphone, we combine the three MPE systems with no empty hypothesis present, the sequence being CTM-A, CTM-B, and CTM-C; following this, we combine the resulting system outputs over the multiple table-top microphones, with the empty hypothesis present. This ROVER arrangement gives us the best results on the development data.

3.4 Language Model

For language modeling, we constructed three separate four-gram models, which were smoothed with modified Kneser-Ney smoothing [12]: One based on 1.5M words of meeting transcript data; a second one using 37M words of scientific conference proceedings (primarily from data processed by CHIL partner LIMSI); and finally, one based on 3M words of Fisher data. To construct the language model (LM) used for the static decoding graph, we interpolated these models with weights of 0.56, 0.37, and 0.07, respectively, and used entropy-based pruning [13] to reduce the resulting model to about 5M n-grams. For the LM employed in lattice rescoring, we used the interpolated models with no pruning.

A 37k-word lexicon was obtained by keeping all words occurring in the meeting transcripts and Fisher data and the 20k most frequent words in the other text corpora. Pronunciations were based on a 45-phone set (41 speech, one silence phone and three noise phones), and were obtained from the Pronlex lexicon, augmented with manual pronunciations.

4 Results and Discussion

As mentioned earlier, we used the CHIL 2005 evaluation data set ("CHIL eval05") as our development set to allow benchmarking progress from last year's CHIL-internal evaluation. We first report development set results, followed by results on the RT06s evaluation set ("CHIL eval06").

4.1 Development Set Results

We first summarize results concerning the acoustic modeling steps discussed in Section 3.2, using the reference segmentation for the MDM condition. Table 4.1 illustrates the gains at each stage, with each word error rate (WER) number obtained by using ROVER on the recognizer output from all four table-top microphones in the CHIL eval05 data (one table-top microphone was omitted due to its very low SNR). The final combined output is obtained by the two-level ROVER process discussed in Section 3.3.

Table 1. Word error rates, %, of MDM systems on the CHIL eval05 dataset, using the reference segmentation

System	Model A	Model B	Model C
SI	55.4	55.4	55.4
MAP-SI	53.1	53.1	53.1
VTLN	—	53.8	55.0
SAT	52.1	51.4	53.4
fMPE+MAP-MPE	46.6	47.5	49.0
final ROVER		45.6	

While Model B provides the best SAT result, Model A gives the best MPE result, with no variance normalization and no VTLN. Discriminative training provides up to 5.5% absolute gain over the SAT system. Applying ROVER on the three MPE systems adds another 1% absolute gain. We see MAP adaptation gaining 2.3% absolute at the initial SI decoding, the output transcripts of which are used for computing the VTLN warping factor and fMLLR transforms. MAP-MPE is necessary after fMPE, for otherwise, MPE after fMPE erases all the gain obtained by fMPE. MAP-MPE gives 0.6% absolute gain on top of fMPE. The reason for MAP-MPE may be that the acoustic conditions of CHIL data are quite different from other meeting data resources.

Table 2. Perplexity and OOV rates of the old LM (used in the CHIL 2005 evaluation) and the newly designed LM

Set	CHIL eval05		CHIL eval06
LM	old LM	new LM	new LM
perplexity	136.7	110.4	119.0
OOV rate (%)	3.9	0.4	1.0

Compared to our last year's LM [3], the new LM incorporates all meeting transcripts available, and a lot more conference proceedings text. Table 2 lists perplexity values and the out-of-vocabulary (OOV) rate of the old vs. new LM on the CHIL eval05 data. Clearly, the new LM results in significantly reduced values of both perplexity and OOV rate. Notice however that the two LMs have drastically different vocabulary sizes of 20k (old) versus 37k (new) words. It is also interesting to note that the new LM generalizes well to the CHIL eval06 data, achieving reasonable perplexity and OOV rate, as depicted in Table 2.

In LM rescoring, silence and noise words are treated as transparent words; while in the static decoding graph, the probabilities of silence and noise are estimated from training data. We optimize the probability value of silence with one table-top microphone in the development data and use it for evaluation data. Table 3 presents the comparison of each table-top microphone before and after the LM rescoring. LM rescoring seems to normalize the text, and the ROVER effect decreases a bit for Model A. The final ROVER result is 0.6% better than

Table 3. Far-field WERs on the CHIL eval05 set using LM rescoring and two-level ROVER over channels and models

	Model A	Model B	Model C
Table Mic	MPE/LM rescoring	MPE/LM rescoring	MPE/LM rescoring
1	51.3 / 48.7	52.0 / 49.6	53.5 / 51.3
2	53.4 / 51.1	54.1 / 52.0	55.0 / 52.7
3	53.5 / 51.7	53.8 / 51.7	55.1 / 52.7
4	53.9 / 51.7	54.4 / 51.9	55.3 / 53.5
ROVER	46.6 / 47.8	47.5 / 46.8	49.0 / 49.0
final ROVER		45.6 / 45.0	

the MPE final ROVER result. Table 3 also demonstrates significant gains from applying ROVER on four table-top microphones (about 4.5% absolute for all three MPE systems).

The models for IHM data are trained as those of Model C. Table 4 presents the gains obtained at each stage of the decoding process using reference segmentation for IHM systems, and the comparison of the old/new LMs. Clearly the new LM gives about 4% absolute gain. MAP-MPE adds 1.4% absolute on top of fMPE, which is more than the gain on MDM data. This is because there exist relatively more CHIL IHM data for MAP adaptation than CHIL far-field data.

The above results are obtained using the reference segmentation. Table 5 shows WERs of Model A on our automatic segmentation of the CHIL eval05 MDM data. Surprisingly, the results are much better (2% – 3% absolute) than those from human segmentation. This may be due to the fact that human transcribers tend to chop a whole sentence for easier processing.

4.2 Evaluation Set Results

Table 6 depicts the WER of the MDM system at various stages of its pipeline, as well as the final system WER, reported on the NIST RT06s lecture meeting data ("CHIL eval06" set), with the overlapping speaker number set to one. Each WER number is obtained by applying ROVER over the table-top microphones of each CHIL site. The final ROVER result is generated by the two ROVER processes discussed in Section 3.3, following the CTM-A, CTM-B, CTM-C sequence.

Table 4. WERs, %, of IHM systems on CHIL eval05 data using reference segmentation

System	old LM	new LM
SI	39.8	35.6
MAP-SI	38.2	34.3
VTLN	38.1	—
SAT	36.9	33.4
fMPE	33.4	29.9
fMPE+MAP-MPE	—	28.5
MLLR	—	26.8
LM rescoring	—	25.4

Table 5. Comparison of WER results, %, using human (reference) versus automatic segmentation on the CHIL eval05 MDM data

system	reference segmentation	automatic segmentation
SI	55.4	53.5
fMPE+MAP-MPE	46.6	43.7

Table 6. MDM system WER on the CHIL eval06 set using automatic segmentation, reported with overlapping speaker number set to one

System	Model A	Model B	Model C
MAP-SI	60.9	60.9	60.9
fMPE+MAP-MPE	51.5	50.6	51.2
ROVER		49.9	
LM rescoring	52.3	50.6	51.4
final ROVER		50.1	
IBM official submission		51.1	

It turns out that Model B produces the best MPE results on the CHIL eval06 data, while Model A is the worst of all three. Furthermore, LM rescoring hurts performance, with the ROVER result on the MPE outputs being 0.2% better (absolute) than the final ROVER system output. Notice also that the IBM official system submission was scored at 51.1% WER, as compared to the 50.1% reported in Table 6. This is due to the fact that, inadvertently, ROVER over the microphone channels was not consistently applied for all CHIL site data. Of course, this inconsistency did not affect the SDM submission, which achieved a 51.4% WER, since only one microphone channel is used in that condition. Nevertheless, the IBM submission for both MDM and SDM conditions scored well in the RT06s STT evaluation on the CHIL seminar (lecture meeting) data. Interestingly, compared to results for CHIL eval05 in Table 3, results for CHIL eval06 are much worse. Inferred from Table 2, the large WER differences between CHIL eval05 and CHIL eval06 sets may be due to acoustic mismatch.

Unfortunately, the IBM submission for the IHM condition has been very unsatisfactory, yielding 52.8% WER. This is mostly attributable to the poor performance of cross-talk removal. Indeed, without applying the cross-talk removal algorithm (i.e., depending on the automatic segmentation alone), the WER gets significantly reduced to 39.5%. Further analysis of the results shows that if all cross-talk was successfully removed, the WER would have been 33.3%; further, if none of the segments were deleted, and assuming that they would have been correctly decoded, the WER would have dropped to 28.7%. It is also interesting to note that decoding the CHIL eval06 IHM data using the *manual* (reference) segmentation (instead of the automatic segmenter) almost halves the WER of the IBM submission to 27.1%. This is of course due to cross-talk elimination and correct speaker segmentation in the manual transcripts, which also positively affect speaker adaptation in the system.

Table 7. Comparison of WER results, %, using manual (reference) versus automatic segmentation (with no additional cross-talk removal) on the CHIL eval06 IHM set. These results are significantly better than the IBM official submission of 52.8% that employed an unsuccessful cross-talk removal algorithm.

segmentation / system	reference (manual)	automatic, with no cross-talk removal
SI	39.1	51.1
fMPE+MAP-MPE	29.5	41.0
MLLR	28.3	40.3
LM rescoring	27.1	39.5

5 Conclusions

We have made significant progress in the automatic transcription of CHIL meeting data. The main difference of our systems from last year is that we built systems from meeting data, instead of adapting existing out-of-domain models to CHIL data. This resulted in 11.5% absolute improvement (from 36.9% to 25.4%) on the close-talking evaluation data of the first CHIL-internal evaluation campaign, used as our development data. However, this progress did not translate into a satisfactory IHM system submission to the RT06s evaluation, due to the failure to successfully remove cross-talk present in the IHM channels. On the other hand, our development efforts paid off in the far-field conditions: Both our MDM and SDM system scored well in the RT06s evaluation campaign on lecture meeting data. Improvements in both acoustic and language modeling led to this success, with ROVER applied across three different acoustic models playing an important role in WER reduction. In particular for the MDM system, this was accompanied by applying ROVER across multiple table-top microphones, as a means of combining the available channels.

Acknowledgements

We wish to thank Karthik Visweswariah and John Hershey for organizing the RT04 and RT05s data corpora for training purposes. We would also like to acknowledge support of this work by the European Commission under integrated project CHIL, "Computers in the Human Interaction Loop", contract number 506909.

References

1. *The LDC Corpus Catalog*, Linguistic Data Consortium, University of Pennsylvania. Philadelphia, PA. [Online]. Available: `http://www.ldc.upenn.edu/Catalog`
2. J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. Wksp. on Automatic Speech Recog. and Understanding (ASRU)*, Santa Barbara, CA, 1997, pp. 347–354.

3. S. Chu, E. Marcheret, and G. Potamianos, "Automatic speech recognition and speech activity detection in the CHIL smart room," in *Proc. Wksp. Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*, Edinburgh, UK, 2005, pp. 332–343.
4. J. Ajmera and C. Wooters, "A robust speaker clustering algorithm," in *Proc. Wksp. on Automatic Speech Recog. and Understanding (ASRU)*, St. Thomas, US Virgin Islands, 2003, pp. 411–416.
5. A. Stolcke, X. Anguera, K. Boakye, O. Cetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng, "Further progress in meeting recognition: the ICSI-SRI Spring 2005 speech-to-text evaluation system," in *Proc. Rich Transcription 2005 Spring Meeting Recog. Eval.*, Edinburgh, UK, 2005, pp. 39–50.
6. M. F. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.
7. G. Saon, G. Zweig, and M. Padmanabhan, "Linear feature space projections for speaker adaptation," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Salt Lake City, UT, 2001, pp. 325–328.
8. S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Atlanta, GA, 1996, pp. 339–341.
9. G. Saon, M. Padmanabhan, and R. Gopinath, "Eliminating inter-speaker variability prior to discriminant transforms," in *Proc. Wksp. on Automatic Speech Recog. and Understanding (ASRU)*, Trento, Italy, 2001, pp. 73–76.
10. D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, vol. 1, Philadelphia, PA, 2005, pp. 961–964.
11. D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Orlando, FL, 2002, pp. 105–108.
12. S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 13, pp. 359–393, 1999.
13. A. Stolcke, "Entropy-based pruning of backoff languge models," in *Proc. DARPA Broadcast News Transcription and Understanding Wksp.*, Lansdowne, VA, 1998, pp. 270–274.

The ICSI-SRI Spring 2006 Meeting Recognition System

Adam Janin[1], Andreas Stolcke[1,2], Xavier Anguera[1,3], Kofi Boakye[1], Özgür Çetin[1], Joe Frankel[1], and Jing Zheng[2]

[1] International Computer Science Institute, Berkeley, CA, U.S.A.
[2] SRI International, Menlo Park, CA, U.S.A.
[3] Technical University of Catalonia, Barcelona, Spain
janin@icsi.berkeley.edu

Abstract. We describe the development of the ICSI-SRI speech recognition system for the National Institute of Standards and Technology (NIST) Spring 2006 Meeting Rich Transcription (RT-06S) evaluation, highlighting improvements made since last year, including improvements to the delay-and-sum algorithm, the nearfield segmenter, language models, posterior-based features, HMM adaptation methods, and adapting to a small amount of new lecture data. Results are reported on RT-05S and RT-06S meeting data. Compared to the RT-05S conference system, we achieved an overall improvement of 4% relative in the MDM and SDM conditions, and 11% relative in the IHM condition. On lecture data, we achieved an overall improvement of 8% relative in the SDM condition, 12% on MDM, 14% on ADM, and 15% on IHM.

1 Introduction

Despite ongoing advances in automatic speech recognition technology, natural multi-person meetings continue to be challenging. The acoustic environment, especially with desktop microphones, is quite variable. Noises such as fans, door slams, and paper rustling all contribute to the acoustic background. Reverberation and echo can also be a significant problem. Typically, recordings from different sites (and even within the same site) use many different types of microphones. Another issue is that meetings contain large amounts of overlap — people end each other's sentences, interrupt, encourage ("uh huh"), laugh, and so on. Finally, the relative paucity of in-domain training data makes it vital to leverage methods and data that have been developed for other genres of speech, such as conversational telephone speech (CTS) and broadcast news (BN).

As for all our recent meeting evaluation systems, our development strategy for RT-06S was to base the system on the SRI-ICSI-UW RT-04F conversational telephone speech recognition system,[1] with improvements incorporated from the previous year's NIST evaluation systems [1,2]. This year, we improved the delay-and-sum algorithm by using a global histogram to discard frames with low correlation and also by using delays selected among the N-best delay scores rather than only the one-best. The nearfield segmenter now uses cross-channel log-energy ratio features (in addition to Mel frequency cepstral coefficients [MFCCs]) integrated directly with a hidden Markov model

[1] As explained later, we also made use of acoustic models developed for BN.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 444–456, 2006.

(HMM) segmenter. Language models were updated by the inclusion of new conference and lecture room transcripts, as well as additional Web data. A new procedure was used to train the phone posterior features, including separate adaptation for both the nearfield and farfield sources (whereas the RT-05S system was adapted only from the nearfield data). HMM adaptation was improved using a data-induced regression class trees (rather than hand-crafted classes). For the lecture room condition, we used a small amount of additional in-domain training data, as well as data from the TED Corpus [3]. Finally, farfield models were trained using both the farfield and nearfield data (instead of just the farfield data).

The evaluation task and data are described in Section 2. Section 3 gives the system description, focusing on new developments relative to the 2005 system [1]. Results and discussion appear in Section 4, followed by conclusions and future work in Section 5.

2 Task and Data

2.1 Test Data

Evaluation data. The RT-06S conference room evaluation data (eval06) consisted of two meetings each from the University of Edinburgh, CMU (Carnegie Mellon University Interactive Systems Laboratory), NIST (National Institute of Standards and Technology), and VT (Virginia Tech), and one meeting from TNO (the Netherlands Organization for Applied Scientific Research). Systems were required to recognize a specific 18-minute segment from each meeting; however, data from the entire meeting was allowed for processing.[2] Separate evaluations were conducted in three conditions:

MDM multiple distant microphones (primary)
IHM individual headset microphones (required contrast)
SDM single distant microphone (optional)

The lecture room data consisted of 120 minutes of seminars recorded by the Computers In the Human Interaction Loop (CHIL) consortium. In addition to the above conditions, lecture data included the following recording conditions:

ADM all distant microphones (optional)
MBF pre-beamformed signal from the Multiple Mark III microphone array (MM3A, optional)

Microphones varied substantially by type and setup, even within each condition. For example, some of the AMI IHM data were recorded with head-mounted lapel microphones, and MDM recording devices ranged from low- and high-quality individual table-top microphones to AMI's circular microphone arrays. Meeting participants included both native and nonnative speakers of English (unlike in CTS evaluations).

[2] We did not find significant gains from adapting on entire meetings, and, except in the acoustic preprocessing, used only the designated meeting excerpts.

Development data. The RT-05S evaluation data were designated as development data for RT-06S, and used by us as an unbiased test set (designated eval05). For the conference room task, the data consisted of ten 12-minute excerpts of meetings from AMI, CMU, ICSI, VT, and NIST. For the lecture room task, the data consisted of 120 minutes of seminars recorded by the CHIL consortium. We also used the same development set as was used in RT-05S [1] for additional tuning.

2.2 Training Data

Training data for the conference room task were identical to that used in RT-05S, and included data from AMI (35 meetings, 16 hours of speech after segmentation), CMU (17 meetings, 11 hours), ICSI (73 meetings, 74 hours), and NIST (15 meetings, 14 hours). The CMU data were of limited use in that only lapel and no distant microphone recordings were available. For the lecture room task, we included the small amount of available CHIL data that were not in the development sets.[3] These data consisted of only the nearfield signals from excerpts of 38 meetings, totaling about 7 hours of speech. We also included the Translingual English Database (TED) [3], using the boom-microphones only and consisting of 39 lectures for about 9 hours worth of speech.

Background training data for the (pre-adaptation) acoustic models consisted of the publicly available CTS and BN corpora. These included about 2300 hours of telephone speech from the Switchboard, CallHome English, and Fisher collections, and about 900 hours of BN data from the Hub-4 and TDT corpora.

3 System Description

3.1 Signal Processing and Segmentation

Distant microphone processing. All distant microphone channels (in both training and test) were Wiener-filtered for noise reduction using a filter developed for the Qualcomm-ICSI-OGI Aurora system [4]. The process was identical to last year [1].

Subsequently, for the ADM and MDM conditions, a delay-and-sum beamforming technique was applied to combine all available distant microphone channels into a single "enhanced" channel. The system is very similar to the one used in the ICSI RT-06S speaker diarization system [5], and is based on last year's system [6] with two main improvements.

The first improvement affects the noise filtering based on the value returned by the generalized cross-correlation algorithm (GCC-PHAT [7]). Frames with a low correlation value indicate increased uncertainty as to whether the returned delay represents the actual TDOA (time delay of arrival). In last year's submission, we filtered out any value smaller than 0.1, assigning the previous nonfiltered delay to such frames (ensuring delay continuity). This caused fewer frames to be filtered in "cleaner" acoustic conditions than in noisy conditions or with worse microphones. This year's submission computed a global histogram of all delays in all channels and determined the threshold so that 10% of frames are dropped.

[3] These data were provided only after the evaluation had started.

Another improvement this year involves the delays selected among the N-best GCC-PHAT computed. At every position, we consider the tradeoff between selecting the main peaks of the GCC-PHAT function and ensuring a continuity on the selected delays in the region surrounding that point. To do so, we apply a two-step Viterbi decoding at two levels. First, at a channel level, we decide which 2-best delays are most probable in each position. Second, at a global level, all combinations of the local 2-best among all channels are considered, and the best combination is chosen. In each step, each possible state has an emission probability equal to the GCC-PHAT value for each delay/combination, and the transition probability between two nodes is inverse proportional to the distance between its delays/combinations, ensuring that the N-best probabilities in a particular instant sum up to 1. We apply a relative weight of 25 to emphasize the transition probabilities.

This newly introduced technique aims to find the optimum tradeoff between reliability (cross-correlation) and stability (distance between contiguous delays). Stability is vital, as our aim is to obtain an optimally improved signal, while avoiding quick changes of the beamforming between acoustic events.

Once the enhanced signal was generated, speech regions were identified using a speech/nonspeech two-class HMM decoder. Resulting segments were combined and padded with silence to satisfy certain duration constraints that had been empirically optimized for recognition accuracy. The algorithm and models were unchanged from last year [1]. Finally, the segments were clustered into acoustically homogeneous partitions, which serve as pseudo-speaker units for normalization and adaptation. The clustering algorithm was also identical to last year's system.

Close-talking microphone processing. The IHM input channels are segmented (without Wiener filtering) into speech and nonspeech regions using an HMM-based speech/nonspeech segmenter [8]. The segmenter is a two-class HMM decoder with each class represented by a three-state phone model. The states are modeled by 256-component multivariate Gaussian mixtures with diagonal covariance matrices. The segmentation proceeds via decoding of the full IHM channel waveform, potentially in a multi-pass fashion with decreased transition penalty between the speech and nonspeech classes. This is done so as to generate segments that do not exceed 60 seconds in length.

The segmenter uses both single- and cross-channel features for speech activity detection. The single-channel features consist of 12th-order Mel-frequency cepstral coefficients, log-energy, and first and second differences. The cross-channel features are maximum and minimum log-energy differences. The log-energy difference represents the log of the ratio of the short-time energy between a given target channel and a nontarget channel. The maximum and minimum values are selected to obtain a fixed number of feature components, given that the number of channels varies between meetings. These cross-channel features are included specifically to address errors caused by cross-channel phenomena such as crosstalk. All features are computed over a window of 25 ms advanced by 20 ms.

A later (i.e., post-evaluation) enhancement to the system consisted of an energy normalization technique being applied prior to computing the log-energy difference features. For a given channel, the minimum frame log-energy of the channel is

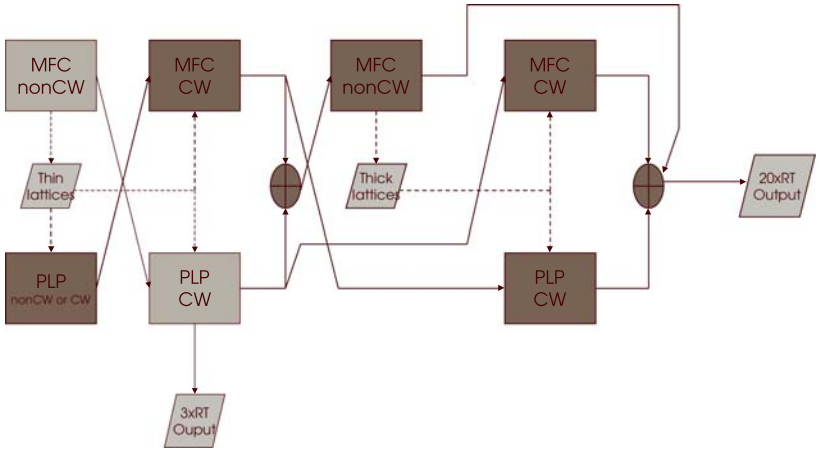


Fig. 1. SRI CTS recognition system. Rectangles represent decoding steps. Parallelograms represent decoding output (lattices or 1-best hypotheses). Solid arrows denote passing of hypotheses for adaptation or output. Dashed lines denote generation or use of word lattices for decoding. Crossed ovals denote confusion network system combination steps.

subtracted from all log-energy values in that channel. That is, for a channel i at frame n

$$E_{norm}(n) \;=\; E_i(n) - E_{min,i} \tag{1}$$

where E represents log-energy. The minimum frame log-energy is used as an estimate of the noise floor and has the advantage of being largely independent of the amount of speech activity in the channel. This normalization was done to compensate for any significant differences in microphone gains and yielded substantial performance improvements over the unnormalized features.

No speaker clustering was performed on the IHM channels, since it was assumed that each IHM channel corresponds to exactly one speaker.

3.2 Acoustic Modeling and Adaptation

Decoding architecture. To motivate the choice of acoustic models, we first describe the SRI-ICSI-UW RT-04F CTS system, on which the meeting system is based (see Figure 1). An "upper" (in the figure) tier of decoding steps is based on MFCC features; a parallel "lower" tier of decoding steps uses perceptual linear prediction (PLP) features. The outputs from these two tiers are combined twice using word confusion networks (denoted by crossed ovals in the figure). Except for the initial decodings, the acoustic models are cross-adapted to the output of a previous step from the respective other tier using maximum likelihood linear regression (MLLR). Lattices are generated initially to speed up subsequent decoding steps. The lattices are regenerated once later to improve their accuracy, after adapting to the outputs of the first combination step. The lattice generation steps use noncrossword (nonCW) triphone models, and decoding from lattices uses crossword (CW) models. Each decoding step generates either lattices

or N-best lists, both of which are rescored with a 4-gram language model (LM); N-best output is also rescored with duration models for words and pauses [9].

The final output is the result of a three-way system combination of MFCC-nonCW, MFCC-CW, and PLP-CW decoding branches. The entire system runs in under 20 times real time (20xRT).[4] The "fast" subset consisting of just two decoding steps (the light-shaded boxes in the figure) runs in about 3xRT; it was used for quick-turnaround experiments, but was not used in this year's evaluation.

Baseline models and test-time adaptation. The MFCC recognition models were derived from gender-dependent CTS models in the RT-04F system, which had been trained with the minimum phone error (MPE) criterion [10] on about 1400 hours of data. (All available native Fisher speakers were used, but to save training time, statistics were collected from only every other utterance). The MFCC models used 12 cepstral coefficients, energy, first-, second-, and third-order differences features, and 2×5 voicing features over a 5-frame window [11]. Cepstral features were computed with vocal tract length normalization (VTLN) and zero-mean and unit variance per speaker/cluster. The 62-component raw feature vector was reduced to 39 dimensions using heteroscedastic linear discriminant analysis (HLDA) [12]. After HLDA, a 25-dimensional Tandem/HATs feature vector estimated by multilayer perceptrons (MLPs) [13,14] was appended. Both within-word and crossword triphone models were trained, for lattice generation and decoding from lattices, respectively. PLP models were based on full-bandwidth analysis, producing 12 coefficients, energy, first-, second- and third-order differences, and then reduced to 39 dimensions using HLDA. (No voicing or MLP features were used in this case.) These models were originally trained on about 900 hours of broadcast news data from the Hub4, TDT2, and TDT4 collections. PLP models are gender-independent. All models were trained using decision-tree-based state tying.

In testing, all models underwent unsupervised adaptation to the test speaker or cluster, using MLLR with multiple, data-induced regression class trees. This is in contrast to last year's system, which used hand-defined MLLR regression classes. The first MFCC and PLP adaptation passes used a phone-loop reference model; later passes adapted to prior recognition output. In addition, all but the first decoding used constrained MLLR in feature space, which was also employed in training (speaker adaptive training) [15].

Acoustic model task adaptation. Nearfield recognition models were adapted to ICSI, NIST, and AMI nearfield microphone data. For the various distant microphone tasks, a single model set was created by adapting to farfield data from ICSI, NIST, and AMI training meetings, plus ICSI, NIST, AMI, and CMU nearfield data. The inclusion of nearfield adaptation data for farfield model training is new this year, and was done with the rationale that farfield recognition involves signals ranging in quality from distant to near-close-talking. Just as in last year's system, we did not delay-sum the training data for the multiple microphone conditions; rather, we pooled all the individual microphone

[4] Runtimes given assume operation with Gaussian shortlists. Since RT-06S did not impose a runtime limit we ran the system without shortlists, in about 25xRT.

signals into one training set, and used the same pooled adaption data for all meetings. The weight for adaptation data statistics was empirically optimized, and set at 20.

Last year, we applied maximum a posteriori adaptation with a maximum mutual information criterion (MMI-MAP) only to the IHM models, and used the standard, less-involved maximum likelihood (ML) MAP procedure on the distant microphone models. This year, MMI-MAP was used for all PLP models, and ML-MAP for all the MFCC+MLP models.

MLP feature adaptation. The MLPs used to estimate Tandem and HATS features were originally trained to perform frame-level phone discrimination using a large subset of the CTS training data [14]. To improve the match to the acoustic conditions of the meeting domain, these were adapted by applying four additional epochs of backpropagation using ICSI, AMI and NIST meeting data as training material. The Karhunen-Loeve transform (KLT) used to reduce the feature dimension from 46 (the size of the phone set) to 25 was kept unchanged from the CTS system, in order to keep the features compatible with existing models. Unlike the ICSI/SRI RT-05S system, in which MLPs were only adapted to nearfield sources, separate MLPs were adapted for the nearfield and farfield conditions. In the case of the HATS, only the merger MLP was adapted, and the 15 critical band networks were left unchanged. The initial learning rates were set to be equal to those at the conclusion of training of the CTS MLPs, and halved after each epoch. The input acoustic parameters used an 8-kHz front end to match that used in the original CTS MLP trainings.

The MLPs for the farfield adaptation were initialized with the nearfield-adapted MLPs after one epoch, and adapted only on regions where there was no overlapping speech. The labels were generated from alignments made on the nearfield data. Initial experiments followed our approach to farfield acoustic model adaptation, in which all available farfield channels were used as training material. Recognition experiments on a development set using these MLPs gave worse performance than using nearfield-adapted MLPs. One possible cause was overtraining, as the MLP was being presented with as many as eight noisy versions of each speech segment during each epoch. We therefore selected a single channel at random to provide the data for each segment (though input normalizations were calculated over all segments for a given speaker/channel combination). This approach led to improved results. Adapting the MLP features to the meeting domain led to reductions in word error rate (WER), in particular for the SDM and MDM conditions, and on the lecture data.

3.3 Language Models

Three LMs were used in decoding: a multiword bigram for lattice generation, a multiword trigram for decoding from lattices, and a word 4-gram for lattice and N-best rescoring. The same set of language models is used for all conference meeting sources (we found no advantage in tuning LMs to the meeting source). A second set of LMs is used for the lecture task.

For the conference room domain, the LMs were linearly interpolated mixtures of component LMs trained from the following sources: (a) Switchboard CTS transcripts, (b) Fisher CTS transcripts, (c) Hub-4 and TDT4 BN transcripts, (d) AMI, CMU, ICSI,

and NIST meeting transcripts, and (e) World Wide Web data newly collected to match different topics and styles, namely, RT-04S meeting sources and AMI meetings, and 525M words of Fisher-like conversational Web data collected and published by the University of Washington for the RT-04F evaluation. The mixture weights were tuned to minimize perplexity on heldout AMI, CMU, ICSI, LDC, and NIST transcripts. The LM vocabulary consisted of 54,524 words, comprising all words in our CTS system (including all Hub-5 and all nonsingleton Fisher words), all words in the ICSI, CMU, and NIST training transcripts, and all nonsingleton words in the AMI training transcripts. The out-of-vocabulary rate was 0.40% on eval04 transcripts, and 0.19% on the 2005 AMI development transcripts.

For the lecture room domain, additional LM mixture components were built from (f) 70K words of CHIL development transcripts and (g) 32M words of speech conference proceedings (suggested by [16]). Also, the Fisher-relevant Web data were replaced by about 512M words of the newly collected Web data related to the CHIL transcripts. The lecture LM mixture was then optimized on CHIL development transcripts. The lecture LM vocabulary was an extension of the conference LM vocabulary, with 3791 additional frequent words found in the proceedings data. The out-of-vocabulary rate on the CHIL development data was 0.18%.

The main difference of this year's conference and lecture LMs compared to last year's is in the Web data LM component. The new Web data collected this year employed a different selection criterion for the n-gram queries submitted to the search engine. Instead of using the most frequent 4-grams in the target corpus, we used the 4-grams with the highest likelihood ratio between a target LM trained on the available meeting or lecture data, and a background LM from all the other data [17]. However, overall, we did not see any significant perplexity or WER improvement over last year's LMs in the eval05 conference test set (the perplexity of the final pruned 4-gram meetings LM was 115); we therefore kept the 2005 LM in our conference evaluation system. On the eval05 lecture task, the new LMs reduced perplexity about 5% relative, to 119, but this improvement did not bring any significant improvement in WER. Nevertheless, the updated lecture LMs were used in the evaluation system, because they were thought to provide better coverage for the new test sets due to the inclusion of more recent Web data, and of the CHIL lecture transcripts.

4 Results and Discussion

Note that all results are reported on non-overlapping speech (using an overlap limit of 1 in the NIST scoring software) in order to be comparable to last year's results.

4.1 IHM Crosstalk Filtering

Table 1 shows IHM recognition results using the eval05 data for the conference room condition. For each row, we show for each meeting recording site the score using unnormalized features and normalized features as described in Section 3 (missing entries were not run for lack of time). We also show the effect of using the SDM channel as a "stand-in" for participants without a microphone. Using the SDM channel does a good job of detecting speech when there is no IHM signal containing the foreground

Table 1. IHM word error using the RT-06S system for the eval05 set on the conference room data with and without energy normalization and with and without the SDM signal

Segmenter Method	Word Error					
	ALL	AMI	CMU	ICSI	NIST	VT
Raw	**25.6**	22.0	23.5	20.9	**37.3**	23.8
Raw + SDM	**24.7**				33.0	
Norm + SDM	**22.7**	21.9	23.1	20.6	**25.2**	22.9
Reference	**19.5**	19.2	19.9	16.8	**21.4**	20.6

Table 2. IHM word error using various segmentation systems on the conference data from the RT-05S and RT-06S evaluations

Segmenter Method	eval05 data	eval06 data
Baseline	29.3	
RT-05S system	25.9	
RT-06S system (raw energies)	24.7	24.0
RT-06S system (normalized energies)	22.7	22.8
Reference	19.5	20.2

speaker. Notice the dramatic improvement in using normalization and the SDM signal for the NIST meeting, in which there was known to be a speaker without microphone (on a speakerphone). The "Reference" row shows the results of a cheating experiment in which the reference segmentation was used. It shows the best we can expect to achieve using an automated method. Though we are approaching this threshold, there is clearly more work that can be done.

The SDM channel was not used in the RT-06S IHM system segmenter, since NIST specified that no un-miced speakers would be present.

Table 2 summarizes the results of the IHM segmenter using various systems on both the eval05 and eval06 evaluation sets.

4.2 Acoustic Modeling

To highlight the improvement in acoustic modeling, Table 3 shows the word error on the SDM and IHM conditions using acoustic models from RT-05S and RT-06S for the conference room and lecture room results. Since IHM and MDM do not use delay-summed signals, these results exclude changes in the delay-sum algorithm. The IHM results were computed using identical segmentations. Also, the language model was kept constant in these experiments. Notice incremental improvements in all conditions, due to the aforementioned improvements in MLP feature training, Gaussian training, and MLLR.

4.3 Result Summary

Table 4 summarizes results on last year's and this year's evaluation sets on the conference room condition. Numbers in parentheses indicate results that were obtained using the new energy normalization technique after the evaluation ended. Relative gains of 3.9% for SDM, 4.0% for MDM, and 6.9% (11.2% post-eval) for IHM were achieved

Table 3. Word error for SDM and IHM conditions on the conference room and lecture room data using 2005 and 2006 acoustic models

Models	RT-05S Conference		RT-05S Lecture	
	SDM	IHM	SDM	IHM
RT-05S	40.9	24.7	51.9	30.8
RT-06S	39.3	24.1	47.4	28.6

Table 4. Word error for conference room data using the 2005 and 2006 systems on the 2005 and 2006 evaluation sets. Numbers in parentheses indicate results obtained after the official evaluation had ended.

System	MDM	SDM	IHM
	eval05 data		
RT-05S system	30.2	40.9	25.9
RT-06S system	29.0	39.3	24.1 (23.0)
	eval06 data		
RT-06S system	34.2	41.2	24.1 (22.8)

on the eval05 data. The difficulty of the eval06 set is comparable to the eval05 set, with the possible exception of the MDM condition, which is slightly worse. We have not yet analyzed this discrepancy.

Table 5 summarizes all results for the lecture room task using the RT-05S and RT-06S systems on the eval05 and eval06 data sets. Notice that eval06 was overall much more difficult than eval05, possibly because of more nonnative speakers, more variation in recording sites, and more channels in the IHM condition (causing more insertion errors from crosstalk).

For all conditions, the RT-06S lecture system shows substantial improvements compared to the RT-05S system, as measured on eval05 data. The gains were 8.1% relative for the SDM condition, 11.9% for MDM, 13.8% for ADM, and 15.0% for IHM.

Looking at lecture recognition results across distant microphone conditions, we see that the delay-sum combination method is effective. Compared to last year's system, this year's system is more robust: last year, the MDM results were worse than the SDM results, whereas the improved delay-sum algorithm now ensures that additional distant microphones always improve results (ADM is better than MDM, which is better than SDM).

Table 5. Word error for the lecture room task using the RT-05S and RT-06S systems on the eval05 and eval06 data sets

System	IHM	SDM	MDM	ADM	UKA/MBF	ICSI/MBF
	eval05 data					
RT-05S system	28.0	51.9	52.0	44.8	-	-
RT-06S system	23.8	47.7	45.8	38.6	-	-
	eval06 data					
RT-06S system	31.0	57.3	55.5	51.0	56.5	56.0

For the MBF (multiple microphone beam-formed) condition, we ran the same system as for the SDM condition, using the beamformed signal as input. For the evaluation, the University of Karlsruhe provided a single beamformed signal based on all the signals from the MM3A microphones [18] (denoted "UKA/MBF" in the table). We were curious to compare the ICSI blind beamformer with the source-localization-based beamformer employed by Karlsruhe, for recognition purposes. Using the same delay-sum procedure as described in Section 3, we generated a new beamformed signal and ran an otherwise identical recognition system ("ICSI/MBF" in the table). Results show that, if anything, the delay-sum method gives somewhat better recognition results. We attribute this to the fact that our algorithm was tuned specifically for recognition accuracy, whereas Karlsruhe's was presumably optimized for source localization accuracy.

Finally, it is interesting to note that the MBF condition performed worse than MDM despite the larger number of microphones (64) in the array. One possible explanation is that the MM3A arrays were located farther from the main speaker than the MDM microphones.

5 Conclusions and Future Work

We continue to make progress in the automatic transcription of conference and lecture room meetings, as measured on NIST evaluation data. Modest gains were achieved in the conference room domain, with the largest improvement coming from the use of integrated cross-channel features in the IHM segmenter. Substantial gains were achieved in the lecture room task through the use of conference-trained distant microphone MLP features, more robust delay-sum, the use of CHIL and TED data to adapt the models, and a small LM improvement. It should be pointed out that all lecture system development occurred within a couple of weeks before and during the evaluation, and further improvements can no doubt be achieved with more careful experimentation.

Plenty of work remains. Several of the system parameters (such as LM weights and insertion penalties) were not properly optimized due to time constraints. Feature mapping techniques could reduce mismatch of the CTS and BN background training data. Given the large number of nonnative speakers of English especially in the lecture data, models adapted to particular accents may improve performance. Finally, although overlapped speech was considered part of the primary condition in this year's evaluation, we made no special effort to handle this type of speech; we consider the detection and modeling of overlapped speech one of the main challenges for future work.

Acknowledgments

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811), and by the Swiss National Science Foundation through NCCR's IM2 project.

Additional support came from the the Defense Advanced Research Projects Agency (DARPA) to SRI under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do

not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC).

We thank José Pardo for contributions to the segmentation and delay-sum algorithms, the members of SRI's Speech Technology and Research Laboratory, as well as Arindam Mandal from the University of Washington for assistance with the recognition system, the U. Washington SSLI laboratory for the computer resources used for web data collection, and all the researchers at ICSI for their help and patience during the evaluation.

References

1. Stolcke, A., Anguera, X., Boakye, K., Çetin, Ö., Grézl, F., Janin, A., Mandal, A., Peskin, B., Wooters, C., Zheng, J.: Further progress in meeting recognition: The ICSI-SRI Spring 2005 speech-to-text evaluation system. In Renals, S., Bengio, S., eds.: Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005. Volume 3869 of Lecture Notes in Computer Science., Springer (2006) 463–475
2. Stolcke, A., Wooters, C., Mirghafori, N., Pirinen, T., Bulyko, I., Gelbart, D., Graciarena, M., Otterson, S., Peskin, B., Ostendorf, M.: Progress in meeting recognition: The ICSI-SRI-UW Spring 2004 evaluation system. In: Proceedings NIST ICASSP 2004 Meeting Recognition Workshop, Montreal, National Institute of Standards and Technology (2004)
3. Lamel, L., Schiel, F., Fourcin, A., Mariani, J., Tillman, H.: The translingual English database (TED). In: Proc. ICSLP, Yokohama (1994) 1795–1798
4. Adami, A., Burget, L., Dupont, S., Garudadri, H., Grezl, F., Hermansky, H., Jain, P., Kajarekar, S., Morgan, N., Sivadas, S.: Qualcomm-ICSI-OGI features for ASR. In Hansen, J.H.L., Pellom, B., eds.: Proc. ICSLP. Volume 1., Denver (2002) 4–7
5. Anguera, X., Wooters, C., Pardo, J.M.: Robust speaker diarization for meetings: ICSI-SRI RT-06S meetings evaluation system. In: Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006. Springer (2007)
6. Anguera, X., Wooters, C., Peskin, B., Aguiló, M.: Robust speaker segmentation for meetings: The ICSI-SRI Spring 2005 diarization system. In Renals, S., Bengio, S., eds.: Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005. Volume 3869 of Lecture Notes in Computer Science., Springer (2006) 402–414
7. Flanagan, J.L., Johnston, J.D., Zahn, R., Elko, G.W.: Computer-steered microphone arrays for sound transduction in large rooms. J. Acoust. Soc. Am. **78** (1985) 1508–1518
8. Boakye, K., Stolcke, A.: Improved speech activity detection using cross-channel features for recognition of multiparty meetings. In: Proc. ICSLP, Pittsburgh, PA (2006)
9. Vergyri, D., Stolcke, A., Gadde, V.R.R., Ferrer, L., Shriberg, E.: Prosodic knowledge sources for automatic speech recognition. In: Proc. ICASSP. Volume 1., Hong Kong (2003) 208–211
10. Povey, D., Woodland, P.C.: Minimum phone error and I-smoothing for improved discriminative training. In: Proc. ICASSP. Volume 1., Orlando, FL (2002) 105–108
11. Graciarena, M., Franco, H., Zheng, J., Vergyri, D., Stolcke, A.: Voicing feature integration in SRI's Decipher LVCSR system. In: Proc. ICASSP. Volume 1., Montreal (2004) 921–924
12. Kumar, N.: Investigation of Silicon-Auditory Models and Generalization of Linear Discriminant Analysis for Improved Speech Recognition. PhD thesis, Johns Hopkins University, Baltimore (1997)
13. Morgan, N., Chen, B.Y., Zhu, Q., Stolcke, A.: TRAPping conversational speech: Extending TRAP/Tandem approaches to conversational telephone speech recognition. In: Proc. ICASSP. Volume 1., Montreal (2004) 536–539
14. Zhu, Q., Stolcke, A., Chen, B.Y., Morgan, N.: Using MLP features in SRI's conversational speech recognition system. In: Proc. Interspeech, Lisbon (2005) 2141–2144

15. Jin, H., Matsoukas, S., Schwartz, R., Kubala, F.: Fast robust inverse transform SAT and multi-stage adaptation. In: Proceedings DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, VA, Morgan Kaufmann (1998) 105–109
16. Lamel, L., Adda, G., Bilinski, E., Gauvain, J.L.: Transcribing lectures and seminars. In: Proc. Interspeech, Lisbon (2005)
17. Wan, V., Hain, T.: Strategies for language model web-data collection. In: Proc. ICASSP. Volume I., Toulouse (2006) 1069–1072
18. Gehrig, T., McDonough, J.: Tracking multiple simultaneous speakers with probabilistic data association filteres. In: Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006. Springer (2007)

The LIMSI RT06s Lecture Transcription System

L. Lamel, E. Bilinski, G. Adda, J.L. Gauvain, and H. Schwenk*

LIMSI-CNRS, BP 133
91403 Orsay Cedex, France

Abstract. This paper describes recent research carried out in the context of the FP6 Integrated Project CHIL in developing a system to automatically transcribe lectures and presentations. Widely available corpora were used to train both the acoustic and language models, since only a small amount of CHIL data was available for system development. Acoustic model training made use of the transcribed portion of the TED corpus of Eurospeech recordings, as well as the ICSI, ISL, and NIST meeting corpora. For language model training, text materials were extracted from a variety of on-line conference proceedings. Experimental results are reported for close-talking and far-field microphones on development and evaluation data.

1 Introduction

One of the CHIL services is to provide on-line and off-line support for lecture situations. For on-line services, the lecture must be transcribed and annotated in close to real time, while the lecture is happening. Such an interactive application would allow latecomers to catch up on what was already presented earlier in the talk, by either reading the transcript or an automatically created summary. If someone needs to step out of the lecture for a few minutes, the service would allow the person to scan the missing portion. Many possible off-line applications can also be envisioned that would benefit from automatic transcription, annotation, indexing and retrieval. These technologies could be used to archive all public presentations (conferences, workshops, lectures and seminars) for future viewing and selected access. Automatic techniques can provide a wealth of annotations, enabling users to search the audio data to find talks on specific topics or by certain speakers. Given the large number of parallel oral sessions at most major conferences, such services could allow attendees to interactively access talks they were unable to attend. At LIMSI our focus is on developing a lecture and seminar transcription system for off-line applications.

The speech recognizer for CHIL has been developed from the LIMSI Broadcast News transcription system for American English [6]. Since only a small amount of CHIL data was available for system development widely available corpora were used to train both the acoustic and the language models. Acoustic model training made use of the transcribed portion of the TED corpus of Eurospeech recordings, the ICSI, ISL, and NIST meeting corpora, and a few CHIL seminars. For language model training,

* This work was partially financed by the European Commission under the FP6 Integrated Project IP 506909 CHIL.

S. Renals, S. Bengio, and J. Fiscus (Eds.): MLMI 2006, LNCS 4299, pp. 457–468, 2006.

in addition to the transcriptions of the audio data, text materials were extracted from a variety of on-line conference proceedings. The LIMSI CHIL speech recognizer used in the January 2005 evaluation is described in [8,10]. In the remainder of this paper the 2006 speech recognizer is described, highlighting differences from the 2005 system and development results are provided.

2 Recognizer Overview

The speech recognizer uses the same core technology and is built using the same training utilities as the LIMSI Broadcast News Transcription system described in [6]. The transcription system has two main components, an audio partitioner and a word recognizer. Data partitioning is based on an audio stream mixture model [6], and serves to divide the continuous stream of acoustic data into homogeneous segments, associating cluster, gender and labels with each non-overlapping segment. This year the data partitioner was adapted to better deal with the farfield microphone data [17]. For each speech segment, the word recognizer determines the sequence of words, associating start and end times and an optional confidence measure with each word. The word recognizer makes use of continuous density HMMs with Gaussian mixture for acoustic modeling and n-gram statistics estimated on large text corpora for language modeling. Each context-dependent phone model is a tied-state left-to-right CD-HMM with Gaussian mixture observation densities where the tied states are obtained via a decision tree.

The language models (LMs) are interpolated backoff n-gram models estimated on subsets of the available training texts. The word list was selected from the audio transcripts and the proceedings texts so as to minimize the out-of-vocabulary (OOV) rate on a set of development data. The 2006 word list is case-sensitive and contains 58k words, and several thousand compound words and acronyms. (The 2005 word list had 35k words, and both the word list and language models were case-insensitive). Pronunciations for several thousand words were added to the LIMSI American English dictionary. Many of the additional words were compound words formed by concatenating pronunciations from existing words, inflected forms and spelled or spoken acronyms.

Word recognition is performed in multiple decoding passes, where each pass generates a word lattice which is expanded with a 4-gram LM. The posterior probabilities of the lattice edges are estimated using the forward-backward algorithm. The 4-gram lattices are converted to a confusion network with posterior probabilities by iteratively merging lattice vertices and splitting lattices edges until a linear graph is obtained. This procedure gives comparable results to the edge clustering algorithm proposed in [13]. The words with the highest posterior in each confusion set are hypothesized. The final decoding pass makes use of a connectionist language model interpolated with a 4-gram model.

3 Training Corpora

Although multi-site data collection is ongoing, only a limited number of transcribed seminars were available for speech recognizer training since priority was given to selection of the evaluation test data. Therefore one of the problems was to locate

Table 1. Summary of audio data sources

Source	*Microphone*	*Type*	*Amount*
TED	lapel	39 lectures	9.3h
ISL	lapel	18 meetings	10.3h
ICSI	head mounted	75 meetings	60h
NIST	head mounted	19 meetings	17.2h
ICSI	tabletop	75 meetings	70h
CHIL	head mounted	17 seminars	6.2h

Table 2. Summary of additional CHIL audio data sources used in 2006. s1 and s2 correspond to different segments from the same seminar.

CHIL_ctm_2003-10-28_[s1,s2], CHIL_ctm_2003-11-11_[s1,s2], CHIL_ctm_2003-11-18_[s1,s2]
CHIL_ctm_2003-11-25_A_[s1,s2], CHIL_ctm_2003-11-25_B_[s1,s2],
CHIL_ctm_2003-12-16_A_[s1,s2], CHIL_ctm_2003-12-16_B_[s1,s2],
CHIL_ctm_20041111_1100, CHIL_ctm_20041111_1400, CHIL_ctm_20041111_1545,
CHIL_ctm_20041112_1030, CHIL_ctm_20041112_1400, ISL_20040614_ctm
ISL_20040616_ctm, ISL_20040621_ctm, ISL_20040721_ctm, ISL_20040830_ctm

appropriate audio and textual resources with which to develop the recognizer models. Of the publicly available corpora, the most closely related audio data are the TED recordings of presentations at the *Eurospeech* conference in Berlin 1993 [9]. The majority of presentations are made by non-native speakers of English. Although there are 188 speeches (about 50 hours) of audio recordings, transcriptions are only available for 39 lectures [1]. Other related data sources are the ISL, ICSI and NIST meeting corpora which contain audio recordings made with multiple microphones of a variety of meetings (3-10 participants) on different topics [3,5,7]. The amount of data per corpus is summarized in Table 1. The first four corpora were used in training the 2005 system, and the last two entries are new in the 2006 system. Using a single microphone channel per speaker for the data from all four sources (distributed by the LDC), a total about 97 hours of audio training data were available in 2005 and an additional 76 hours of data were used in 2006. As can be seen, in the 2005 system only close-talking microphone data were used for acoustic model training, whereas some distant microphone channels were used in 2006.

Since one of the aims in CHIL is speech recognition of farfield data, and the primary RT06s task being the multiple distant microphone condition, in the 2006 system farfield data were also used in training. To this end a selection of the farfield data in the ICSI corpus were used. Since for the ISCI data there are a varying number of channels, for each of the speakers, a single farfield channel was selected as being the most appropriate for that speaker. The microphone channel was chosen as that having the highest likelihood during forced alignment on a subset of data for each speaker. During training these data were pooled with the close-talking microphone data. The CHIL seminars included in this year's training are listed in Table 2.

The language model training data consist of manual transcriptions of related audio data as well as the proceedings texts from a variety of speech and language related

Table 3. Summary of audio transcripts

TED presentations: 71k words
NIST meetings: 156k words
ISL meetings: 116k words
ICSI: 785k words
CTS: 3M words
AMI/IDIAP meeting: 143k words
NIST RT04, RT05 data: 57k words
CHIL Jun04/Jan05 seminars: 55k words
CHIL summer04 seminars: 38k words

Table 4. Summary of proceedings texts (20k articles, 42M words)

TED texts:	426 papers	929k words
ASRU'99-05:	427 papers	1140k words
DARPA'97-99,04:	119 papers	317k words
Eurospeech'97-05:	3485 papers	7650k words
ICASSP'95-05:	7831 papers	14318k words
ICME'00,03:	996 papers	2101k words
ICSLP'96-04:	3202 papers	7198k words
LREC'02,04:	891 papers	2553k words
ISCA+other workshops:	2333 papers	6077k words

conferences and workshops. The audio transcripts come from the same sources as are used for acoustic training. In addition transcriptions of conversational telephone speech from the CallHome, SwitchBoard and Fisher collections (distributed by the LDC) were used. We also tried using assorted transcriptions from Broadcast News (BN) data, but since these did not reduce the perplexity they were not used to estimate the language models. The amount of words in the each audio transcript source are given in Table 3. Compared with last year's system we made use of some additional data from the AMI/IDIAP meeting corpus, the NIST RT04 and RT05 development data, and the transcripts of the additional CHIL seminars (24k words more than last year).

In addition to the audio transcripts, a large number of texts on audio, speech and language processing can be obtained from conference and workshop proceedings as shown in Table 4. The almost 20k papers in the proceedings texts were processed using scripts derived from ones shared by ITC-IRST to convert postscript and pdf files to ascii texts. The texts were extracted from the PDF files, after removing corrupted documents. Further processing removed unwanted materials (email, websites, telephone numbers, addresses, mathematical formulas and symbols, figures, tables, references) as well as special formatting characters and ill-formed lines. A stricter filtering was applied this year than last year in order to have cleaner texts for language model training.

4 Acoustic Modeling

The acoustic feature vector has 39-components comprised of 12 cepstrum coefficients and the log energy, along with the first and second order derivatives. The cepstral para-

meters are derived from a Mel frequency spectrum estimated on the 0-8kHz band every 10ms. For each 30ms frame the Mel scale power spectrum is computed, and the cubic root taken followed by an inverse Fourier transform. Then LPC-based cepstrum coefficients are computed. The cepstral coefficients are normalized on a segment-cluster basis using cepstral mean removal and variance normalization. Thus each cepstral coefficient for each cluster has a zero mean and unity variance.

The acoustic models are context-dependent, 3-state left-to-right hidden Markov models with Gaussian mixture. The triphone-based phone models are word-independent and gender-independent, but word position-dependent. The acoustic models are MLLT-SAT trained, with different sets of tied-state models are used in successive decoding passes. State-tying is carried out via divisive decision tree clustering, constructing one tree for each state position of each phone so as to maximize the likelihood of the training data using single Gaussian state models, penalized by the number of tied-states [6]. A set of 152 questions concern the phone position, the distinctive features (and identities) of the phone and the neighboring phones.

The baseline acoustic models were those used in the LIMSI 2005 CHIL system. This year several enhancements were added to the system. Speaker adaptive (SAT) training was used in the system which gave about a 1% absolute improvement for the ihm condition. The acoustic models were trained on multi-style data, including close-talking and farfield microphones. Different training strategies were investigated, and the best for us was to pool the close-talking head-mounted data with the table-top audio data for the models used in the ihm condition. For the farfield conditions (sdm, mdm, and mm3a) the best results were obtained by adapting the ihm models with the farfield data. We also explored adapting broadcast news models since this was advantageous last year, but better results were obtained on the development data using lecture models.

5 Language Modeling

The 58k recognizer word list was determined using the following process: The 75k most probable words were selected by linear interpolation of the unigram language models obtained from the 8 different seminar and meeting data sources listed in Table 3 and a component trained on 46M words of proceedings texts. While the CTS data were used for language model training, they were not used for vocabulary selection. Compared with last year, about 300k words of additional audio transcripts were available. This 75k word list was then filtered using our master pronunciation dictionary for American English in order to eliminate errors. (Pronunciations for several hundred frequently occurring words in the transcripts had already been added to the master dictionary for AM training.) The final word list contains 57768 words, and has an OOV rate of 0.46% on the development data (RT05s eval). Last year's 35k word list had an OOV rate of 0.61% on the same data.

For language model estimation the available corpora were grouped into 3 sources:

- Seminar and meeting transcriptions (1.42M words)
- Proceedings texts (46M words)
- Transcriptions of Conversational Telephone Speech databases available from LDC (29M words)

The proceeding texts are comprised of the proceedings from 54 conferences and workshops in speech and language, which represent about 20,000 PDF documents.

Three backoff n-gram language models were estimated, one on each of the data subsets. The component language models were interpolated [16], and the weights were chosen to minimize the perplexity of the development data. The largest weight is for the transcriptions (0.6), with weights of 0.3 and 0.1 for the proceedings texts and CTS transcripts respectively. The perplexity of the 4-gram LM is 130 on the development data, which can be compared to 140 with last years' model.

Our text normalization process has been drastically changed since last year. Our previous normalization was case insensitive (all words were in upper case), no compound words were allowed, and all acronyms were split in a sequence of letters. In the new normalization, some compound words are kept (words which contain an internal hyphen such as *'air-conditioning'*) according to a reference list. The reference list of compound word was derived from compound words in the American Heritage Dictionary, completed with a list of first and last names in different languages, and some place names found in geographical and historical encyclopedias. The texts were also processed with a primitive named entity detector, in order to preserve the proper names.

Thus, for instance, for a compound word 'A-B' present in the text, 2 cases are distinguished:

1. If the word 'A-B' is present in one of the reference lists or was tagged as a 'proper name', the word 'A-B' is kept as a lexical entry.
2. If the first case is not true, the word 'A-B' is split into the two words 'A B'.

The new text processing is case sensitive, which means that to have correct texts, a decision must be taken as to what is the true case for the first word of each sentence. Moreover, in some texts word case may be vague either for stylistic reasons (signifying emphasis) or due to segmentation errors in the proceedings texts. For these texts the case of all words needs to be reconsidered. This process is done by adding all the possible cases encountered in the texts for all words for which case is potentially ambiguous. For this process the available 472M words of broadcast news texts were also used. To attribute the correct case for the sentence-initial word an interpolated language model was constructed with a set of texts after removing the first word of each sentence. Case is then added to the original sentence by creating a graph with all possible cases for all words with multiple forms, and parsing the graph using the interpolated language model. Finally, all acronyms are considered as words and have not been split.

The aim of this new normalization is not to optimize the lexical coverage or to decrease the perplexity of the language model, but to be able to deliver a transcription containing more information, and thus facilitate the further downstream processing (named entity detection, summarization, ...).

Connectioninst language models [2,14] have been shown to be performant when LM training data is limited. The basic idea is to project the word indices onto a continuous space and to use a probability estimator operating on this space, as illustrated in Figure 1. Both tasks are performed by a neural network. This is still a n-gram approach, but the n-gram LM probabilities are "interpolated" for any possible context of length n-1 instead of backing-off to shorter contexts. Since the resulting probability densities

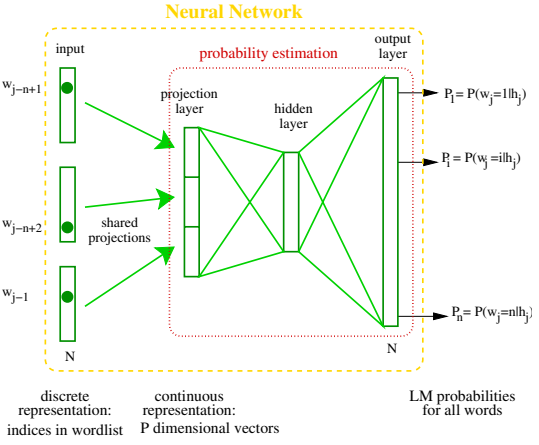


Fig. 1. Neural network language model

are continuous functions of the word representation, better generalization to unknown n-grams can be expected.

The neural network LM was trained on the transcriptions of the audio data and the proceedings, but not the CTS data. This language model reduces the perplexity on the development data from 135 to 108.

5.1 Decoding

Word recognition is performed in two passes, where each decoding pass generates a word lattice which is expanded with a 4-gram LM. The posterior probabilities of the lattice edges are estimated using the forward-backward algorithm. The 4-gram lattices are converted to a confusion network with posterior probabilities by iteratively merging lattice vertices and splitting lattices edges until a linear graph is obtained. This procedure gives comparable results to the edge clustering algorithm proposed in [13]. The words with the highest posterior in each confusion set are hypothesized.

Pass 1: Initial Hypothesis Generation - This step generates initial hypotheses which are then used for speaker-based acoustic model adaptation. This is done via one pass (about 1xRT) cross-word trigram decoding with gender-independent sets of position-dependent triphones (5k contexts, 5k tied states) and a 35k word trigram language model (15M trigrams and 4M bigrams). The trigram lattices are rescored with a 4-gram language model (7M fourgrams, 15M trigrams and 4M bigrams).

Pass 2: Adapted decode - Unsupervised acoustic model adaptation of speaker-independent models is performed for each speaker using the CMLLR and MLLR techniques [11] with only two regression class. The lattice is generated for each segment using a 58k word bigram LM and position-dependent triphones with 24k contexts and 11k tied states (32 Gaussians per state). As in the first pass, the lattices are rescored with a 58k word 4-gram language model (9M fourgrams, 19M trigrams and 5M bigrams).

6 Experiments and Results

All the development work at LIMSI was carried out using a portion of the RT05s evaluation data for which audio files and transcripts were shared by UKA. The data consist of the ihm channels of excerpts from 16 seminars shown in Table 5. For the sdm condition and mdm ccondition the subset of seminars used are listed in the lower part of the table.

Table 5. Development set used at LIMSI for ihm and sdm/mdm conditions (from RT05s eval)

ihm:
CHIL_20041123-0900-[E1,E2]_h01_001, CHIL_20041123-1000-[E1,E2]_h01_001,
CHIL_20041123-1100-[E1,E2,E3]_h01_001, CHIL_20041123-1500-[E1,E2]_h01_001,
CHIL_20041123-1600-[E1,E2]_h01_001, CHIL_20041124-1000-[E1,E2]_h01_001,
CHIL_20041124-1100-[E1,E2]_h01_001, CHIL_20050112-0000-[E1,E2]_h01_001,
CHIL_20050126-0000-E1_h01_001, CHIL_20050127-0000-E1_h01_001,
CHIL_20050128-0000-[E1,E2]_h01_001, CHIL_20050202-0000-[E1,E2]_h01_001,
CHIL_20050214-0000-E1_h01_001, CHIL_20050310-0000-[E1,E2]_h01_001,
CHIL_20050310-0001-E1_h01_001, CHIL_20050314-0000-[E1,E2]_h01_001
sdm/mdm:
CHIL_20041123-1600-E1: d01, d02, d03,d04, d05 CHIL_20050202-0000-E2: d01, d02, d03,d04, d05
CHIL_20050314-0000-E2: d01, d02, d03,d04, d05 CHIL_20050128-0000-E1: d01, d02, d03,d04, d05
CHIL_20050310-0001-E1: d01, d02, d03,d04, d05

The starting point was an updated version of the LIMSI 2005 evaluation system which incorporated an automatic data partitioner (last year's evaluation used manually determined speech segments). As cen be seen in Table 6 the word error rate with the baseline system was 26.1%. This is the same system as was used to automatically transcribe the Q&A development seminars. Using updated acoustic models (AM 1), with the 35k LM from last year gave a small increase in performance. Resgmenting the training data and reestimating acoustic models (AM 2) gave an additional small gain, with a larger gain from SAT training (AM 2 + SAT). The use of the 58k wordlist and language model with the baseline acoustic models gave essentially the same performance as the baseline system (verifying that the new normalization did not degrade performance). Retraining acoustic models with the same text normalization results in a better match of the cross-word context-dependent phone models during training and test, as can be seen in the second entry in the lower part of Table 6. Tuning the system gave a further improvement, as did speaker adaptive training (+SAT), pronunciation probabilities (+ pron probs) and the connectionist language model (+ NNLM). The neural network LM achieves almost a 1% absolute word error reduction on top of the other improvements. Overall on the development data a relative word error reduction of 13% was obtained compared to the 2005 system.

The close-talking microphone contrast condition for RT06s aimed at transcribing only the speech from the primary talker. This is very different from the previous CHIL evaluations where the segments of speech from other speakers were removed prior to scoring. In order to compare the performance of the LIMSI 2006 system to that of the 2005 system, we decided to score using the same method as was used last year, that is ignoring speech in inter-segment gaps. (Strong arguments can be made for both points

Table 6. Word error rates on the ihm development data (see Table 5)

System	*WER (%)*
Baseline, 35k LM	26.1
Updated AM 1, 35k LM	25.9
Updated AM 2, 35k LM	25.7
Updated AM 2+SAT, 35k LM	25.0
58k wordlist, LM	26.0
58k LM, updated AM	25.3
+ tuning	24.6
+ SAT	24.0
+ pron probs	23.5
+ NNLM	22.6

of view. At LIMSI we believe that the role of the speech recognizer is to transcribe speech into words, independent of who spoke them. It is the role of the speaker diarization system to associate words with the person who spoke them. This is not the point of view taken in the RT06s evaluation, and the evaluation plan, while clearly stating this for SAD is a bit ambiguous on this point for STT.) The LIMSI system did not make any attempt to exclude speech from other speakers and therefore has a very high insertion rate (over 121%, more than 22k words), giving an overall error rate of 147%. (Note that the development set we used did not contain background speech so it was not possible to do a serious development activity. Therefore we did not try.)

Table 7 gives unofficial, comparative results on the RT06s evaluation data with the baseline system and the RT06s evaluation system on the ihm audio data, ignoring speech in inter-segment gaps. Although these numbers cannot be compared to other sites, they allow us to measure the improvement of our models this year. The overall error rate has been reduced by almost 12% absolute (28% relative). A more appropriate measure would be to score all of the speech recognized, but at the time of this writing reference transcriptions for the inter-segment gap regions are not available.

Table 7. Unofficial, comparative results on the RT06s evaluation data with the baseline system and the RT06s evaluation system on the ihm audio data, ignoring speech in inter-segment gaps. There are a total of 19373 reference words.

System	*Cor*	*Subs*	*Del*	*Ins*	*WER*
ihm baseline	64.0	25.7	10.3	6.1	42.1
ihm RT06s	72.9	19.5	7.6	3.2	30.3

Development for the farfield data condition was carried out on portion of the available data consisting of 5 seminars listed in the lower portion of Table 5. Initially the standard data partitioner was used, but when the SAD/SPKR systems for RT06s were finalized (see [17]), these were used for further development work and in the final evaluation system. As can be seen in Table 8, the word error on the sdm data was reduced from 64% to 55% when scoring with overlap. The mdm system output was created by

combining with ROVER [4] the outputs of all the sdm channels. For this condition the development word error rate was reduced from 55.7% to about 51%.

Table 8. Word error rates on the sdm and mdm development data (see Table 5)

	sdm WER (%)		*mdm WER (%)*
System	*overlap*	*non overlap*	*overlap*
58k LM, update AM 2	64.0	62.9	
58k LM, adapt AM with FF	62.5	61.3	55.7
+ tuning	60.4	58.8	
+ SAT	60.1	58.5	
+ mdm partitioner	56.6	57.1	53.3
+ pron prob	56.1	55.3	
+ NNLM	55.2	54.4	51.1

Table 9. NIST official RT06s results on farfield data: mdm and sdm conditions (top). Unofficial, comparative results on the RT06s evaluation beam-formed data distributed by UKA with the baseline farfield models and a post RT06 model set adapted with 5 hours of beam-formed data. There are a total of 17986 reference words. (Results are missing for seminar AIT_20051011_B_Segment1 due to a partitioning error for the baseline system.)

System	*Cor*	*Subs*	*Del*	*Ins*	*WER*
mdm	48.3	39.3	12.4	11.8	63.5
sdm	43.2	38.6	18.2	6.5	63.3
mm3a baseline	32.6	20.8	46.6	1.4	68.8
mm3a post RT06	48.1	31.7	20.2	5.6	57.5

The first two entries in Table 9 give the official NIST results for the for the mdm and sdm conditions. As for the development results, the mdm hypotheses are obtained by applying ROVER [4] to the hypotheses of the individual microphone channels. The lower part of the table gives unofficial contrastive results on the mm3a beam-formed multiple mark III microphone array data provided by UKA for the baseline RT06s farfield system and with acoustic model adaptation using about 5 hours of beam-formed multiple mark III microphone array data also provided by UKA. These data correspond to the portion of the development data for which manual transcripts were available. Since no data remained for system development, we chose not to submit a system for this condition in the official evaluation. Although all the word error rates are quite high, the best performance is obtained on the mm3a data using the acoustic models adapted with the beam-formed data.

7 Conclusions

This paper has described our research aimed at developing a system to automatically transcribe lectures and seminars for off-line applications. Publicly available corpora

were used to train both the acoustic and language models, since only a small amount of CHIL data were available for system development. Results were reported for both close talking and far-field microphones, for both development and evaluation data. This was LIMSI's first participation to the multiple farfield microphone task. Compared to the LIMSI 2005 system, the overall error rate has been reduced by over 10% on the development data for the ihm condition and about 15% on the RT06s eval mm3a data.

References

1. The Translanguage English Database (TED) Transcripts, LDC catalog number LDC2002T03, isbn 1-58563-202-3.
2. Y. Bengio and R. Ducharme, "A neural probabilistic language model," *Advances in Neural Information Processing Systems (NIPS)*, **13**:933-938, 2001.
3. S. Burger, V. MacLaran and H. Yu, "The ISL Meeting Corpus: The Impact of Meeting Type on Speech Style, *ICSLP'02*, Denver, Sep 2002. (LDC2004S05, LDC2004E04, LDC2004E05)
4. J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," *Proc. ASRU'97*, 347-354, Santa Barbara, December 1997.
5. J.S. Garofolo, C.D. Laprun, M. Michel, V.M. Stanford and E. Tabassi, "The NIST Meeting Room Pilot Corpus," *LREC'04*, Lisbon, May 2004. (LDC2004S09, LDC2004T13)
6. J.L. Gauvain, L. Lamel, G. Adda, "The LIMSI Broadcast News Transcription System," *Speech Communication*, **37**(1-2):89-108, May 2002.
7. A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, C. Wooters, "The ICSI Meeting Corpus," *ICASSP'03*, Hong Kong, Apr 2003. (LDC2004S02, LDC2004T04)
8. L. Lamel, G. Adda, E. Bilinski and J.L. Gauvain, "Transcribing Lectures and Seminars," *Proc. ISCA Eurospeech'05*, Lisbon, Sep 2005.
9. L.F. Lamel, F. Schiel, A. Fourcin, J. Mariani and H. Tillmann, "The Translanguage English Database TED," *ICSLP'94*, Yokohama, Sep 1994. (LDC2002S04)
10. L. Lamel, H. Schwenk, J.L. Gauvain, G. Adda and E. Bilinski, "Improvements in Transcribing Lectures and Seminars," *Proc. MLMI'05*, Edinburgh, July 2005.
11. C.J. Leggetter, P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, **9**(2):171-185, 1995.
12. D. Macho, J. Padrell, A. Abad, C. Nadeu, J. Hernando, J. McDonough, M. W olfel, U. Klee, M. Omologo, A. Brutti, P. Svaizer, G. Potamianos, and S. Chu, "First experiments of automatic speech activity detection, source localization and speech recognition in the CHIL project," *Workshop on Hands-Free Speech Communication and Microphone Arrays*, Rutgers University, Piscataway, NJ, 2005.
13. L. Mangu, E. Brill and A. Stolcke, "Finding Consensus Among Words: Lattice-Based Word Error Minimization," *Eurospeech'99*, 495-498, Budapest, Sep 1999.
14. H. Schwenk, "Efficient training of large neural networks for language modeling," *IJCNN* , pp. 3059–3062, 2004.
15. A. Waibel, H. Steusloff, R. Stiefelhagen, "CHIL - Computers in the Human Interaction Loop," *5th International Workshop on Image Analysis for Multimedia Interactive Services*, Lisbon, April 2004. (http://isl.ira.uka.de/chil)

16. P.C. Woodland, T. Niesler and E. Whittaker, "Language Modeling in the HTK Hub5 LVCSR," presented at the 1998 Hub5E Workshop, Sep 1998.
17. X. Zhu, C. Barras, L. Lamel and J-L. Gauvain, "Speaker Diarization: from Broadcast News to Lectures," *Proc. RT06s*, submitted.
18. X. Zhu, C. Barras, S. Meignier and J.L. Gauvain, "Combining speaker identification and BIC for speaker diarization" *Proc. Interspeech'05, pp.2441-2444*, Lisboa, September, 2005
19. X. Zhu, C.C. Leung, C. Barras, L. Lamel, and J.L. Gauvain, "Speech activity detection and speaker identification for CHIL," *Proc. MLMI'05*, Edinburgh, July 2005.

Author Index