LNCS 2997

Sharon McDonald
John Tait (Eds.)

Advances in Information Retrieval

26th European Conference on IR Research, ECIR 2004
Sunderland, UK, April 2004
Proceedings

Springer

Lecture Notes in Computer Science 2997
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Sharon McDonald John Tait (Eds.)

Advances in Information Retrieval

26th European Conference on IR Research, ECIR 2004
Sunderland, UK, April 5-7, 2004
Proceedings

Springer

eBook ISBN: 3-540-24752-1
Print ISBN: 3-540-21382-1

Created in the United States of America

Preface

These proceedings contain the refereed full technical papers presented at the 26th Annual European Conference on Information Retrieval (ECIR 2004). ECIR is the annual conference of the British Computer Society's specialist group in Information Retrieval. This year the conference was held at the School of Computing and Technology at the University of Sunderland ECIR began life as the Annual Colloquium on Information Retrieval Research. The colloquium was held in the UK each year until 1998 when the event was held in Grenoble, France. Since then the conference venue has alternated between the United Kingdom and Continental Europe, and the event was renamed the European Conference on Information Retrieval. In recent years, ECIR has continued to grow and has become the major European forum for the discussion of research in the field of Information Retrieval. To mark this metamorphosis from a small informal colloquium to a major event in the IR research calendar, the BCS-IRSG decided to rename the event to the European Conference on Information Retrieval.

ECIR 2004 received 88 full paper submissions, from across Europe and further afield including North America, China and Australia, a testament to the growing popularity and reputation of the conference. Out of the 88 submitted papers, 28 were accepted for presentation. All papers were reviewed by at least three reviewers. Among the accepted papers 11 have a student as the primary author, illustrating that the traditional student focus of the original colloquium is alive today.

The collection of papers presented in this book reflect a broad range of IR problems. Contributions from keynote speakers Gary Marchionini and Yorick Wilks kick start the proceedings with Marchionini's proposal for a new paradigm for IR, based on his emphasis on the interactive nature of IR tasks, and Wilks' thought provoking discussion of the role of NLP techniques in IR. The organization of the proceedings reflects the session structure of the conference, topics covered include user interaction, question answering, information models, classification, summarization, image retrieval, evaluation issues, cross language IR and categorization, summarization, information models, question answering, cross language IR, image retrieval and Web-based and XML retrieval.

I am indebted to many individuals for the quality of this year's conference proceedings. Specifically, I would like to acknowledge the significant efforts of the programme committee, my co-chair John Tait and posters chair Michael Oakes. Thank you for your hard work, and for meeting the tight deadlines imposed. It has been my pleasure to work with you to produce a high-quality conference programme. Thanks also to the conference gold sponsors, Microsoft Research, Canon UK, Leighton Internet, BCS-IRSG, and the University of Sunderland.

Finally, I would like to extend my thanks to Arthur Wyvill and John Cartledge for their work on the paper submission system, Zia Syed for his help in publicizing ECIR 2004 and Lesley Jenkins for her excellent administrative support. Most of all, I would like to thank my husband Alan Lumsden for his love and support as well as the invaluable contribution he made at various stages in the development of ECIR 2004.

January 2004 Sharon McDonald

Organization

ECIR 2004 was organized by the School of Computing and Technology, University of Sunderland, United Kingdom.

Programme Committee

Sharon McDonald, University of Sunderland, United Kingdom (Chair)
John Tait, University of Sunderland, United Kingdom (Chair)
Michael Oakes, University of Sunderland, United Kingdom (Posters Chair)

Andrew MacFarlane, City University, United Kingdom
Alan Smeaton, Dublin City University, Ireland
Alessandro Sperduti, University of Padova, Italy
Ali Asghar Shiri, University of Strathclyde, United Kingdom
Andreas Rauber, Vienna University of Technology, Austria.
Ari Pirkola, University of Tampere, Finland
Arjen de Vries, CWI, Netherlands
Avi Arampatzis, University of Utrecht, Netherlands
Ayse Göker, Robert Gordon University, United Kingdom
Barry Smyth, University College Dublin, Ireland
Chris Mellish, University of Aberdeen, United Kingdom
Claudio Carpineto, Fondazione Ugo Bordoni, Italy
David Harper, Robert Gordon University, United Kingdom
David Losada, University de Santiago de Compostela, Spain
Djoerd Hiemstra, University of Twente, Netherlands
Dunja Mladenić, Jožef Stefan Institute, Slovenia
Fabio Crestani, University of Strathclyde, United Kingdom
Fabrizio Sebastiani, National Council of Research, Italy
Gabriella Pasi, National Council of Research, Italy
Gareth Jones, Dublin City University, Ireland
Giambattista Amati, Fondazione Ugo Bordoni, Italy
Giuseppe Amato, National Council of Research, Italy
Gloria Bordogna, CNR, IDPA, Italy
Iadh Ounis, University of Glasgow, United Kingdom
Ian Ruthven, University of Strathclyde, United Kingdom
Ion Androutsopoulos, Athens University of Economics and Business, Greece
Jane Reid, Queen Mary, University of London, United Kingdom
Jesper Wiborg Schneider, Royal School of Library and Information Science, Denmark
Joemon Jose, University of Glasgow, United Kingdom
Johannes Füernkrantz, Austrian Research Institute for Artificial Intelligence, Austria

Josiane Mothe, University Paul Sabatier, France
Jussi Karlgren, Swedish Institute of Computer Science, Sweden
Kees Koster, University of Nijmegen, Netherlands
Keith van Rijsbergen, University of Glasgow, United Kingdom
Leif Azzopardi, University of Paisley, United Kingdom
Marcello Federico, ITC-irst, Italy
Margaret Graham, Northumbria University, United Kingdom
Mark Girolami, University of Glasgow, United Kingdom
Marko Grobelnik, Jožef Stefan Institute, Slovenia
Massimo Melucci, University of Padova, Italy
Micheline Beaulieu, University of Sheffield, United Kingdom
Mohand Boughanem, University Paul Sabatier, France
Monica Landoni, University of Strathclyde, United Kingdom
Mounia Lalmas, Queen Mary, University of London, United Kingdom
Nicholas Kushmerick, University College Dublin, Ireland
Norbert Fuhr, University of Duisburg-Essen, Germany
Pasquale Savino, National Council of Research, Italy
Patrick Gallinari, University of Paris, France
Peter Ingwersen, Royal School of Library and Information Science, Denmark
Pia Borlund, Royal School of Library and Information Science, Denmark
Ricardo Baeza-Yates, Univeristy of Chile, Chile
Robert Gaizauskas, Univeristy of Sheffield, United Kingdom
Sándor Dominich, University of Veszprém, Hungary
Tony Rose, Cancer Research, United Kingdom
Umberto Straccia, National Council of Research, Italy
Wessel Kraaij, TNO TPD, Netherlands
Yoelle Maarek, IBM Research, Isreal
Yves Chiaramella, Joseph Fourier University, France

Best Student Paper Award Committee

Sándor Dominich, University of Veszprém, Hungary (Chair)
Giambattista Amati, Fondazione Ugo Bordoni, Italy
Pia Borlund, Royal School of Library and Information Science, Denmark

Additional Reviewers

Anastasios Tombros, Queen Mary, University of London, United Kingdom
Christopher Stokoe, University of Sunderland, United Kingdom
Gilles Hubert, IRIT, France
Janez Brank, Jožef Stefan Institute, Slovenia
Theodora Tsikrika, Queen Mary, University of London, United Kingdom

Sponsoring Institutions

Microsoft Research

Leighton
Innovation as standard

Canon

Canon
Research Centre
Europe

University of
Sunderland

INFORMATION RETRIEVAL
SPECIALIST GROUP
BCS

This page intentionally left blank

Table of Contents

Keynote Papers

User Studies

Question Answering

Information Models

Classification

Summarization

Image Retrieval

Evaluation Issues

Cross Language IR

Web-Based and XML IR

This page intentionally left blank

From Information Retrieval to Information Interaction

Gary Marchionini

University of North Carolina at Chapel Hill, School of Information and Library Science
100 Manning Hall
Chapel Hill, NC 27599, USA
march@ils.unc.edu

Abstract. This paper argues that a new paradigm for information retrieval has evolved that incorporates human attention and mental effort and takes advantage of new types of information objects and relationships that have emerged in the WWW environment. One aspect of this new model is attention to highly interactive user interfaces that engage people directly and actively in information seeking. Two examples of these kinds of interfaces are described.

1 Introduction

Information retrieval (IR) is hot. After 40 years of systematic research and development, often ignored by the public, technology and a global information economy have conspired to make IR a crucial element of the emerging cyberinfrastrucure and a field of interest for the best and brightest students. The new exciting employers are Google, Amazon, and eBay and the extant giants like IBM and Microsoft have active IR research and development groups. In many ways, research in IR had plateaued until the WWW breathed new life into it by supporting a global marketplace of electronic information exchange. In fact, I argue that the IR problem itself has fundamentally changed and a new paradigm of information interaction has emerged. This argument is made in two parts: first, the evolution of IR will be considered by a broad look at today's information environment and trends in IR research and development and second, examples of attempts to address IR as an interactive process that engages human attention and mental effort will be given.

2 Information Objects and People

As a scientific area, IR uses analysis to break down the whole problem into components and first focus on the components that promise to yield to our techniques. IR has always been fundamentally concerned with information objects and with the people who create, find, and use those objects; however, because people are less predictable and more difficult and expensive to manipulate experimentally, IR research logically focused on the information objects first. Traditionally, information objects have been taken to be documents and queries and research has centered on two basic issues: representation of those objects and definition of the relationships

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 1–11, 2004.

among them. Representation is a classical issue in philosophy, information science (e.g., Heilprin argued that compression was the central representation problem [9]), and artificial intelligence. The IR community has demonstrated a variety of effective representations for documents and queries, including linguistic (e.g., controlled vocabulary) assignments and a large variety of mathematical assignments (e.g., vectors) based on term-occurrence, relevance probability estimates, and more recently hyperlink graphs. IR research has mainly focused on equality (e.g., of index terms) and similarity relationships—similarity between/among objects—and developed a large variety of matching algorithms that are exploited in today's retrieval systems. A schematic for the traditional IR problem is depicted in Figure 1.

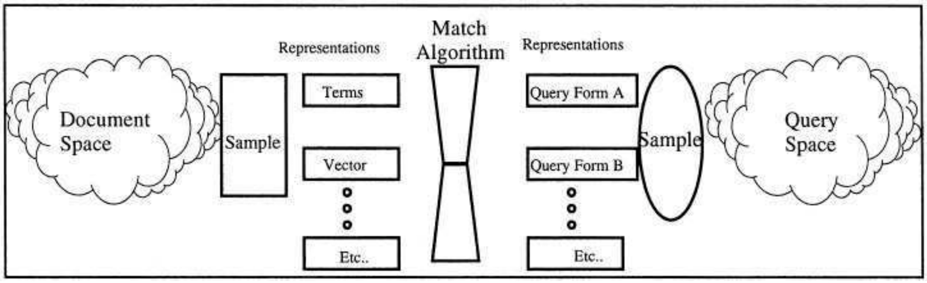


Fig. 1. Content-Centered Retrieval as Matching Document Representations to Query Representations

The figure shows that samples of document and query objects from the respective universe of all objects are each represented in some fashion, most often using the same representation form. For example, a simple approach used in early commercial retrieval systems was to represent documents and queries with terms assigned from a controlled vocabulary and simply match overlaps. A more contemporary example returns ranked sets of similarities by representing documents and queries as vectors of inverse document frequency values for a specific set of terms in the sample ordered by cosine similarity. In cases where the document and query representations are in different forms (e.g., different metadata schemes or human languages), crosswalks, translations, or interlingua must also be added to the process. This content-centered paradigm has driven creative work and led to mainly effective retrieval systems (e.g., SMART, Okapi, Iquery), however, progress toward improving both recall and precision seems to have reached a diminishing return state.

Two important changes have been taking place in the electronic information environment that expand this schema and stimulate new kinds of IR research and development. These changes are due to new types and properties of information objects and to increasing attention to human participation in the IR process. The IR community has begun to recognize these changes as illustrated by the two grand research and development challenges identified for IR research at a recent strategic workshop [1]: global information access ("Satisfy human information needs through natural, efficient interaction with an automated system that leverages world-wide structured and unstructured data in any language."), and contextual retrieval ("Combine search technologies and knowledge about query and user context into a single framework in order to provide the most 'appropriate' answer for a user's information needs." P.330).

The very nature of information objects of interest to IR has both broadened and qualitatively morphed. On one hand, IR has broadened its interest in objects that are not strictly text to include statistics, scientific data sets and sequences, images, sounds, video, animations and other multimedia. In many cases, the same retrieval paradigms are applied with these kinds of objects (e.g., color histograms rather than term occurrences). Additionally, new kinds of objects are emerging such as executable code modules, transaction protocols and forms, and advisory agents and processes, each offering new kinds of feature sets that may be leveraged for retrieval. What is more significant than new types of objects is the trend toward all objects becoming more dynamic and less static and dependable for IR purposes. For example, an active blog is an object that is continually changing and its representations must likewise be continually updated as well. This change emanates from new capabilities within objects and from new capabilities in the external environment that contains them. Internally, electronic objects increasingly are designed to exhibit behavior—to ‘act’ according to external conditions. Hypertext is of course the classic example; recommender systems illustrate more contemporary examples; and context-aware sensor-program devices illustrate the latest trend. In addition to the increasingly sophisticated behavior inherent to information objects is the trend toward the global information space (cyberinfrastructure) to store and use context. For example, a retrieval system may not only represent webpage content but continuously update access times and referral pages. Additionally, the system may save increasingly detailed traces of fleeting ephemeral states arising in online transactions—perhaps as extreme as client-side mouse movements as well as clicks. Thus, our objects acquire histories, annotations, and linkages that may strongly influence retrieval and use. It is important to keep in mind that this applies to query objects as much as document objects. For example, consider the implications for retrieval of a query on the World Trade Center before and after 9/11.

These changes in the very nature of information objects offer new challenges and opportunities for IR. The IR community has moved to accept these challenges on multiple fronts—consider for example, the evolution of the TREC tracks over time. Clearly, entirely new kinds of features are available to use in our object representations. Likewise, object contexts will help enormously in representation and revealing relationships. What seem particularly promising are opportunities to discover new kinds of features within objects, and new kinds of relationships among objects that can be leveraged for retrieval purposes. Hyperlinks and citations are literal relationships formed by object creators and these relationships have been creatively used as features for representing those objects in page rank and hub-authority algorithms. Explicit recommendations are relationships formed by third parties between objects and opinion and can be used to cluster objects of similar opinion. Implicit recommendations are relationships formed by social behavior—the traces of many people acting with objects–and are also leveraged for retrieval purposes. I suspect there are scores of features natively associated with electronic objects (e.g., trustworthiness, various costs, perceptibility) and even more relationships among electronic objects (e.g., counterargument, derived from, alternative for) that have yet to be tapped for retrieval purposes. The truly exciting thing about IR today is that there is so much new ground to plow that even relative novices can make important discoveries.

Taken alone, this basic change in the nature of information would bloat the basic IR paradigm with a large array of alternative representation options and matching algorithms. A second trend has been in play that combines to require a new paradigm for IR. This trend is an increasing consideration of the people using an information retrieval system. Although there have always been voices representing people in IR research (e.g., advocates of subjective relevance such as Saracevic [17], Schamber [18], and Harter [7]; those who focused on the cognitive processes in retrieval such as Belkin [2], Ingwersen [10], and Marchionini [12]), there are increasing efforts in the IR research community to incorporate people into the retrieval problem. This represents maturation in our approach to IR research and development as we aim to expand our problem-definition to include major facets that have long been set aside in order to focus on the content facets of the overall IR problem.

Figure 2 depicts a different paradigm for the retrieval problem than the classic matching paradigm in Figure 1. The information sample here is shown as a cloud rather than a fixed database since it is dynamic. In this figure, the emphasis is on the flow of representations and actions rather man discrete matches. The indexes are multiple and dynamic. The classical techniques for representing information objects remain useful but may be controlled/selected by users rather than fixed by the system. The relationships of similarity, however, may be determined by the human information seeker on the fly according to their needs and capabilities. Thus, the problem shifts from the system optimizing matching to put burden on the human information seeker to engage in an ongoing process. In such a user-centered paradigm, people have responsibilities and capabilities. Expecting a two-word query to Google to solve every information need is both lazy and naïve and people must go beyond this to be successful. One challenge is that people tend to want to be lazy and naïve (this is sometimes a good cost-benefit tradeoff decision) when doing complex and tedious tasks, especially in the many cases when retrieval tasks are embedded in larger activities. Our approach to this challenge is to imagine information seeking as a core life process where people are constantly connected to information sources just as our bodies are connected to the environment through filters and selectors that are highly tuned to the environment. In such a paradigm, the crucial system design challenges become the control mechanisms for interacting with representations in agile and engaging ways. Note that some of these interactions incorporate the existing query/results patterns so ubiquitous today.

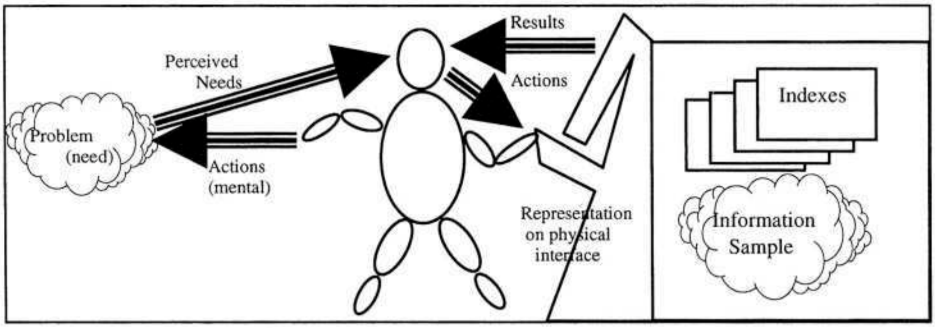


Fig. 2. User-centered information interaction

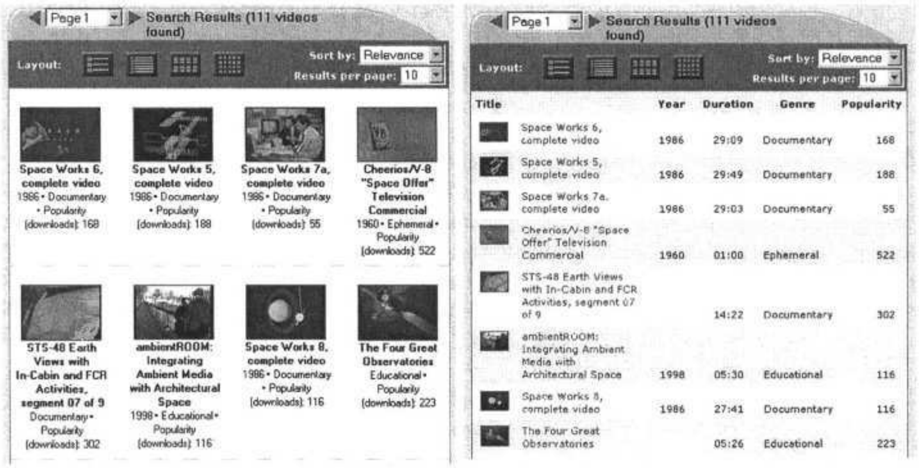
IR research brings users to bear in various ways. There have been long-standing efforts to provide automatic or user-controlled query expansion [e.g., 14], systems that strongly benefit from user relevance feedback [e.g., 15, 16], and efforts to provide user assistance [e.g., 13]. In addition to leveraging human-generated metadata, researchers are looking for ways to use user behavior or conscious annotations to add additional metadata/features to objects. Some have worked toward modeling users with profiles that may be explicitly completed by users or automatically generated by monitoring actions. More recently, attention to recommender systems both explicitly and automatically capitalizes on user actions with information systems [e.g., 11]). These efforts are even being subsequently leveraged for opinion mining that generates new inferred relationships that may in turn be used as features for retrieval. Other efforts aim toward providing memory aids for users (e.g., [5]) and in extreme examples, of gathering more or less complete interaction histories [e.g., 3]. Other approaches may leverage caching to preserve user-system interaction states over long periods of time (the Internet Archive preserves web page states, but imagine resources for preserving all interactions with specific resources for very long periods—something telephone companies have had to do routinely). Still others aim to create integrated environments that use data mining rules and contemporaneous activity to contextual information retrieval [e.g., 4]. There are likewise aims to create anticipatory information systems that go well beyond the selective dissemination systems of yore to leverage context and user profiles. All of these efforts will enrich a human-centered IR paradigm and advance the field toward more complete consideration of all information seeking factors. I estimate that the greatest progress come from methods that actively include human capabilities in the IR process. To this end, a number of researchers aim to focus on the crucial human-system interaction mechanism that serves as the linchpin of this paradigm.

3 Highly Interactive Interfaces

The concept of direct manipulation was introduced into interface design by Shneiderman [20] and has been applied to interfaces specific to information retrieval by a number of researchers. Shneiderman and his colleagues applied direct manipulation techniques in retrieval environments called dynamic query systems [19]. The key elements of these highly interactive interfaces are active engagement of the user with visual representations that update immediately upon input actions and allow immediate reversal of actions. In the case of dynamic queries, there is close coupling of results display and mouse or keyboard actions. Other researchers have created highly interactive environments for database and information retrieval, most notably, the Xerox PARC group's series of systems (e.g.., the Hyperbolic Tree, Perspective Wall, Web Forager). See Hearst [8] for a review of interfaces for IR. Two examples from our work at Carolina on agile view interfaces for information interaction follow.

3.1 Digital Video Retrieval

As part of our efforts to develop an open source video digital library (www.open-video.org). we have created and systematically evaluated a series of visual surrogates (representations) for video content. These surrogates include keyframe-based storyboards and slide shows, fast forwards, and excerpts. These surrogates are considered to be alternative view options for users who have identified a partition of the corpus (either through a text query or a set of interactions/selections with corpus overview representations). Figure 3 shows an Open Video Overview for a set of videos. The views are agile in that simple mouse actions are used to change views. Users can choose which representation best meets their needs and quickly get overviews of the result set with different visual and textual cues emphasized. Clicking on a specific surrogate for a segment (see Figure 4), brings up a full metadata record with three different previews specific to that video's content: a 7 second excerpt, a storyboard (containing up to 36 keyframes), and a fast forward (at 64X speed). Clicking one of the radio buttons immediately displays the preview in the preview panel without opening new windows or changing the user's context. Textual metadata is also displayed. The retrieval paradigm is to allow the user to move quickly to different levels of video granularity, most of which have alternative representations that emphasize different video features, in order to both determine whether it is worth loading and viewing the video and to understand the overall context within which the specific video sits. The actual system also leverages other relationships such as popularity of download, similarity based on usage within a session (implicit recommendations), and various standard bibliographic indexes with hyperlinks as appropriate. Note that textual queries are also supported—the idea is not to replace a demonstrably useful capability but to augment it and put information seekers in control of strategic search decisions.



Space Works 6, complete video
1986 • Documentary • Popularity (downloads): 168

Space Works 5, complete video
1986 • Documentary • Popularity (downloads): 188

Space Works 7a, complete video
1986 • Documentary • Popularity (downloads): 55

Cheerios/V-8 "Space Offer" Television Commercial
1960 • Ephemeral • Popularity (downloads): 522

STS-48 Earth Views with In-Cabin and FCR Activities, segment 07 of 9
Documentary • Popularity (downloads): 302

ambientROOM: Integrating Ambient Media with Architectural Space
1998 • Educational • Popularity (downloads): 116

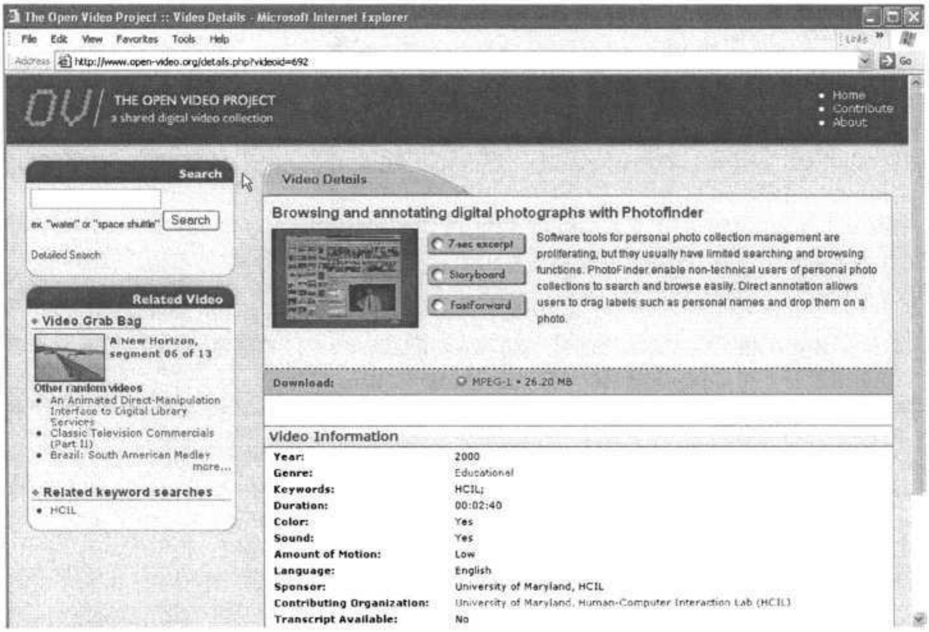
Space Works 8, complete video
1986 • Documentary • Popularity (downloads): 116

The Four Great Observatories
Educational • Popularity (downloads): 223

Title	Year	Duration	Genre	Popularity
Space Works 6, complete video	1986	29:09	Documentary	168
Space Works 5, complete video	1986	29:49	Documentary	188
Space Works 7a. complete video	1986	29:03	Documentary	55
Cheerios/V-8 "Space Offer" Television Commercial	1960	01:00	Ephemeral	522
STS-48 Earth Views with In-Cabin and FCR Activities, segment 07 of 9		14:22	Documentary	302
ambientROOM: Integrating Ambient Media with Architectural Space	1998	05:30	Educational	116
Space Works 8, complete video	1986	27:41	Documentary	116
The Four Great Observatories		05:26	Educational	223

Fig. 3. Alternative Overviews of a Search Result Set

What I consider most significant about this project is that the design decisions are based on an interaction framework (agile views) and the object representations are empirically validated with extensive user testing (see the project website for a series of papers). This principled and systematic approach to system development is rooted in the strong evaluation cultures of the HCI and IR communities. User feedback and strong growth in usage of the Open Video digital library demonstrates the efficacy of this empirical evaluation tradition to theory and development.


The Open Video Project :: Video Details - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.open-video.org/details.php?videoid=692
THE OPEN VIDEO PROJECT
a shared digital video collection
Home
Contribute
About
Search
ex. "water" or "space shuttle"
Search
Detailed Search
Related Video
Video Grab Bag
A New Horizon, segment 06 of 13
Other random videos
An Animated Direct-Manipulation Interface to Digital Library Services
Classic Television Commercials (Part II)
Brazil: South American Medley
more...
Related keyword searches
HCIL
Video Details
○ 7-sec excerpt
○ Storyboard
○ FastForward
Download:
MPEG-1 • 26.20 MB

Browsing and annotating digital photographs with Photofinder

Software tools for personal photo collection management are proliferating, but they usually have limited searching and browsing functions. PhotoFinder enable non-technical users of personal photo collections to search and browse easily. Direct annotation allows users to drag labels such as personal names and drop them on a photo.

Video Information

Year:	2000
Genre:	Educational
Keywords:	HCIL;
Duration:	00:02:40
Color:	Yes
Sound:	Yes
Amount of Motion:	Low
Language:	English
Sponsor:	University of Maryland, HCIL
Contributing Organization:	University of Maryland, Human-Computer Interaction Lab (HCIL)
Transcript Available:	No

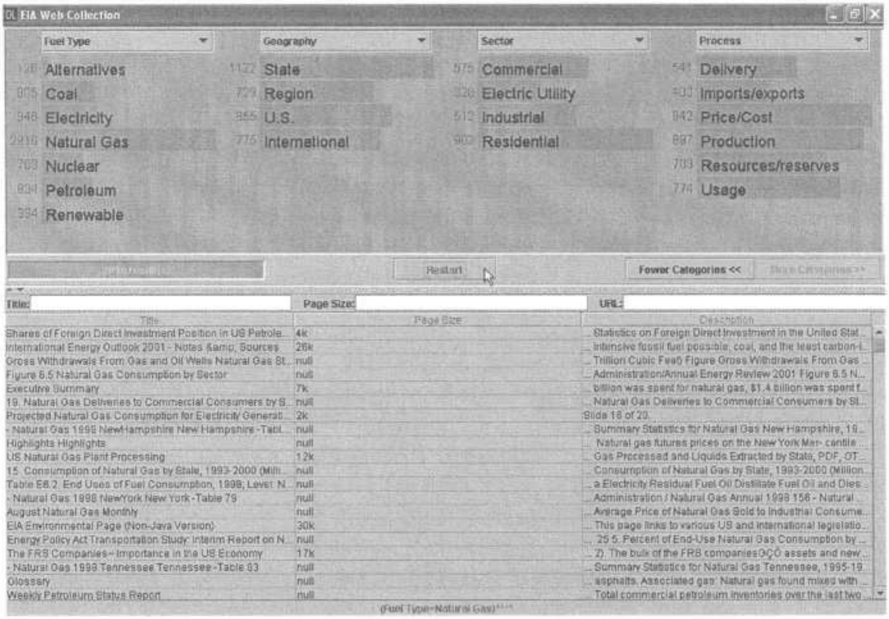
Fig. 4. Alternative Previews for a Specific Video Segment

3.2 Interacting with Databases of Webpages

Another example of a highly interactive interface that couples query and results seamlessly is the Relation Browser++ (http://idl.ils.unc.edu/rave). The idea is to present users with a complete information space and allow them to interact with a variety of partitions based upon several attribute sets. Figures 5-7 show a sequence of explorations in the US Energy Information Administration website with more than 10000 webpages represented in the underlying database. The webpages have been classified according to their pertinence to four main facets (these facets and the classifications shown here are based on the EIA website structure): fuel types, geography, sector, and process, each of which has a small number of attribute values (sub-facets). The opening screen (not shown) provides the number of pages and relative-length bars for each attribute value in the entire corpus. Note that pages can be indexed by several attribute values within and across facets. As the user moves the mouse over an attribute value, the number of pages and bars in other facet attribute values are immediately updated. This allows people to explore relationships across facets. Clicking on an attribute value partitions the database to include only those webpages meeting that condition. Browsing/mousing can continue dynamically, or the user can click the search button to retrieve the results. In Figure 5, the user has clicked on the attribute value 'natural gas' and the search button. We see that 2916 pages are related to natural gas, 128 of these are also related to alternative fuels, 576 are related to the commercial sector, and 403 are related to imports/exports. The search button changes to 'restart' after clicking and the results are displayed in a panel in the same window. This is an important element of interactive interfaces—

maintaining user context so that interactive flow is not lost. New window displays should be avoided until discrete transition points. In the RB++, all browse and search activity takes place in the same window with updates optimized to avoid cognitive interrupts. New windows are only used when the user clicks on a record to jump to that particular webpage. Note that the number of results appears to the left of the search button and the SQL-like query statement that produced the results appears at the bottom of the window.

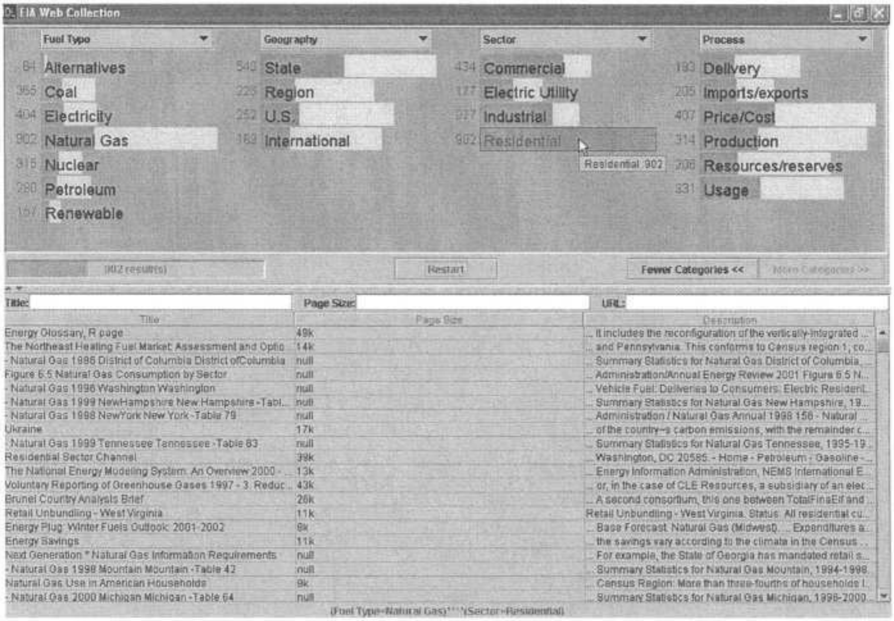
At this point, the user can continue browsing, execute a string search within the results, retrieve a specific page, or start over. Figure 6 shows an example where the user simply moved the mouse to the residential attribute value. All the attribute value numbers and bars are immediately updated; thus 902 webpages are related to natural gas and the residential sector, and 540 webpages are related to residential sector natural gas at the state level. Moreover, note that the posting results have also been updated to show 902 results are available. Each mouse move is in fact a new query. The user can continue to narrow down the results by mousing or can enter text queries in any of the three result set fields (title, page size, URL). The string search immediately returns matches anywhere in the field with the matching characters highlighted. Figure 7 shows the results at the instant that the 's' in the string 'house' is typed from the state shown in Figure 5—yielding 50 results from the 2916 natural gas pages.



Title	Page Size	Description
Shares of Foreign Direct Investment Position in US Petrole...	4k	...Statistics on Foreign Direct Investment in the United Stat...
International Energy Outlook 2001 - Notes &amp; Sources	26k	...intensive fossil fuel possible, coal, and the least carbon-i...
Gross Withdrawals From Gas and Oil Wells Natural Gas St...	null	...Trillion Cubic Feet) Figure Gross Withdrawals From Gas ...
Figure 6.5 Natural Gas Consumption by Sector	null	...Administration/Annual Energy Review 2001 Figure 6.5 N...
Executive Summary	7k	...billion was spent for natural gas, $1.4 billion was spent f...
19. Natural Gas Deliveries to Commercial Consumers by S...	null	...Natural Gas Deliveries to Commercial Consumers by St...
Projected Natural Gas Consumption for Electricity Generati...	2k	Slide 18 of 20.
- Natural Gas 1998 NewHampshire New Hampshire -Tabl...	null	...Summary Statistics for Natural Gas New Hampshire, 19...
Highlights Highlights	null	... Natural gas futures prices on the New York Mer- cantile ...
US Natural Gas Plant Processing	12k	...Gas Processed and Liquids Extracted by State, PDF, OT...
15. Consumption of Natural Gas by State, 1993-2000 (Milli...	null	...Consumption of Natural Gas by State, 1993-2000 (Million...
Table E8.2. End Uses of Fuel Consumption, 1998; Level: N...	null	...a Electricity Residual Fuel Oil Distillate Fuel Oil and Dies...
- Natural Gas 1998 NewYork New York -Table 79	null	...Administration / Natural Gas Annual 1998 156 - Natural ...
August Natural Gas Monthly	null	...Average Price of Natural Gas Sold to Industrial Consume...
EIA Environmental Page (Non-Java Version)	30k	...This page links to various US and international legislatio...
Energy Policy Act Transportation Study: Interim Report on N...	null	... 25.5. Percent of End-Use Natural Gas Consumption by ...
The FRS Companies-- Importance in the US Economy	17k	...2). The bulk of the FRS companiesÕÇÖ assets and new ...
- Natural Gas 1998 Tennessee Tennessee -Table 03	null	...Summary Statistics for Natural Gas Tennessee, 1995-19...
Glossary	null	...asphalts. Associated gas: Natural gas found mixed with ...
Weekly Petroleum Status Report	null	...Total commercial petroleum inventories over the last two ...

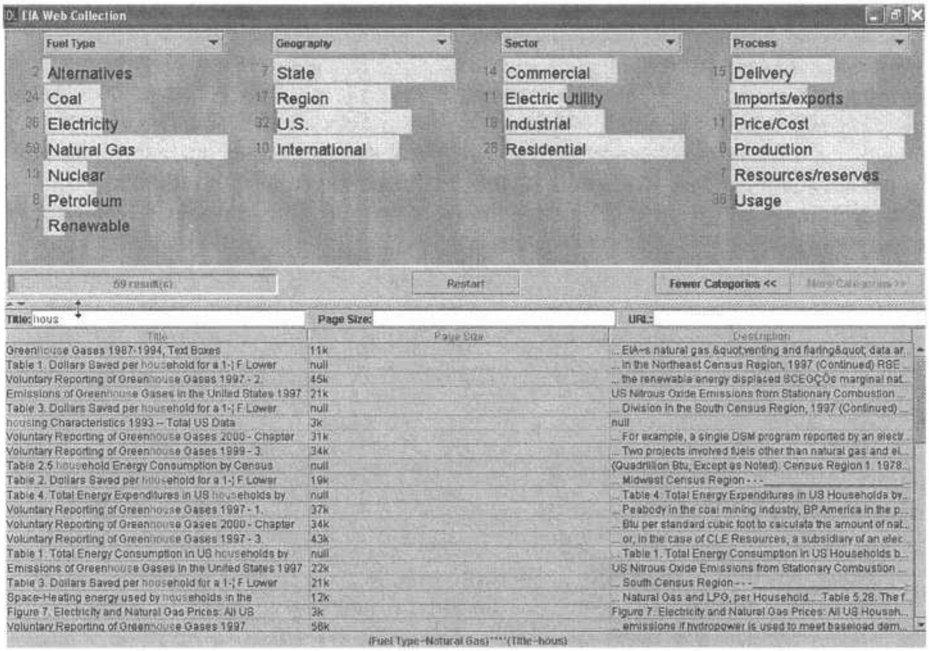
(Fuel Type=Natural Gas)

Fig. 5. Relation Browser++ display for Energy Information Administration websites after clicking on natural gas.



Title	Page Size	Description
Energy Glossary, R page	49k	... It includes the reconfiguration of the vertically-integrated ...
The Northeast Heating Fuel Market: Assessment and Optio	14k	... and Pennsylvania. This conforms to Census region 1, co...
- Natural Gas 1996 District of Columbia District ofColumbia	null	... Summary Statistics for Natural Gas District of Columbia, ...
Figure 6.5 Natural Gas Consumption by Sector	null	... Administration/Annual Energy Review 2001 Figure 6.5 N...
- Natural Gas 1996 Washington Washington	null	... Vehicle Fuel: Deliveries to Consumers: Electric Resident...
- Natural Gas 1999 NewHampshire New Hampshire -Tabl...	null	... Summary Statistics for Natural Gas New Hampshire, 19...
- Natural Gas 1998 NewYork New York -Table 79	null	... Administration / Natural Gas Annual 1998 156 - Natural ...
Ukraine	17k	... of the country~s carbon emissions, with the remainder c...
- Natural Gas 1999 Tennessee Tennessee -Table 83	null	... Summary Statistics for Natural Gas Tennessee, 1995-19...
Residential Sector Channel	39k	... Washington, DC 20585. - Home - Petroleum - Gasoline -...
The National Energy Modeling System: An Overview 2000 - ...	13k	... Energy Information Administration, NEMS International E...
Voluntary Reporting of Greenhouse Gases 1997 - 3. Reduc...	43k	... or, in the case of CLE Resources, a subsidiary of an elec...
Brunei Country Analysis Brief	26k	... A second consortium, this one between TotalFinaElf and ...
Retail Unbundling - West Virginia	11k	Retail Unbundling - West Virginia. Status: All residential cu...
Energy Plug: Winter Fuels Outlook: 2001-2002	9k	... Base Forecast. Natural Gas (Midwest). ... Expenditures a...
Energy Savings	11k	... the savings vary according to the climate in the Census ...
Next Generation * Natural Gas Information Requirements	null	... For example, the State of Georgia has mandated retail s...
- Natural Gas 1998 Mountain Mountain -Table 42	null	... Summary Statistics for Natural Gas Mountain, 1994-1998.
Natural Gas Use in American Households	9k	... Census Region: More than three-fourths of households l...
- Natural Gas 2000 Michigan Michigan -Table 64	null	... Summary Statistics for Natural Gas Michigan, 1996-2000...

Fig. 6. Relation Browser++ display after moving mouse over residential attribute



Title	Page Size	Description
Greenhouse Gases 1987-1994, Text Boxes	11k	... EIA~s natural gas &quot;venting and flaring&quot; data ar...
Table 1. Dollars Saved per household for a 1-¦ F Lower	null	... in the Northeast Census Region, 1997 (Continued) RSE ...
Voluntary Reporting of Greenhouse Gases 1997 - 2.	45k	... the renewable energy displaced ßCEGÇÖs marginal nat...
Emissions of Greenhouse Gases in the United States 1997	21k	US Nitrous Oxide Emissions from Stationary Combustion ...
Table 3. Dollars Saved per household for a 1-¦ F Lower	null	... Division in the South Census Region, 1997 (Continued) ...
housing Characteristics 1993 -- Total US Data	3k	null
Voluntary Reporting of Greenhouse Gases 2000 - Chapter	31k	... For example, a single DSM program reported by an electr...
Voluntary Reporting of Greenhouse Gases 1999 - 3.	34k	... Two projects involved fuels other than natural gas and el...
Table 2.5 household Energy Consumption by Census	null	(Quadrillion Btu, Except as Noted). Census Region 1. 1978...
Table 2. Dollars Saved per household for a 1-¦ F Lower	19k	... Midwest Census Region - - - ________________
Table 4. Total Energy Expenditures in US households by	null	... Table 4. Total Energy Expenditures in US Households by...
Voluntary Reporting of Greenhouse Gases 1997 - 1.	37k	... Peabody in the coal mining industry, BP America in the p...
Voluntary Reporting of Greenhouse Gases 2000 - Chapter	34k	... Btu per standard cubic foot to calculate the amount of nat...
Voluntary Reporting of Greenhouse Gases 1997 - 3.	43k	... or, in the case of CLE Resources, a subsidiary of an elec...
Table 1. Total Energy Consumption in US households by	null	... Table 1. Total Energy Consumption in US Households b...
Emissions of Greenhouse Gases in the United States 1997	22k	US Nitrous Oxide Emissions from Stationary Combustion ...
Table 3. Dollars Saved per household for a 1-¦ F Lower	21k	... South Census Region - - - ________________
Space-Heating energy used by households in the	12k	... Natural Gas and LPG, per Household.... Table 5.28. The f...
Figure 7. Electricity and Natural Gas Prices: All US	3k	Figure 7. Electricity and Natural Gas Prices: All US Househ...
Voluntary Reporting of Greenhouse Gases 1997	58k	... emissions if hydropower is used to meet baseload dem...

Fig. 7. Relation Browser++ display after clicking on residential and entering first four characters of word ‘house’ in the title field.

The Relation Browser++ is the third iteration in an ongoing effort to develop a general purpose interface that couples browsing and searching in databases of web content. The current version is a Java applet linked to a MySQL database. The Relation Browser++ and its predecessors (RAVE-the relation attribute viewer, and original relation browser) have been applied to dozens of applications (see examples at http://idl.ils.unc.edu/rave) where the number of facets and the number of attribute values are small (to preserve screen real estate to ensure no scrolling in the browsing panel). The interface works with medium size databases of tens of thousands of records. We have an implementation with almost 3 million transaction log records but getting the metadata to the client-side applet is very slow. For very large databases like a WWW search, the Relation Browser+ would be most appropriate as a way to interact with a set of search results or a specific category of pages. Our current work aims to develop automatic ways to find good slicing facets and then populate the underlying databases with appropriate pages. The first problem is the category discovery problem and the second is the text classification problem. We are investigating clustering techniques with some customized heuristics [6] with several of our statistical agency research partners.

4 Conclusion

The examples given here are meant to illustrate beginning steps toward engaging people in a continuous way during information seeking. The video retrieval and database browsing examples aim to harvest the fruits of IR progress so that machines can do what they do best under the control of humans with capabilities to recognize patterns and draw inferences from a wide array of possible relationships. In this way, information is created on the fly as well as retrieved (e.g., as users see new patterns or relationships among objects). A person with an information problem is best able to meet that need through action, perception, and reflection rather than through query statements alone. Thus, the notion of information interaction rather than information retrieval to better reflect the active roles of people and the dynamic nature of information objects in the electronic environment.

Acknowledgements. This work has been partially supported by National Science Foundation Grants # IIS 0099638 and EIA 0131824. Thanks are also due to Sheila Denn and Miles Efron who provided thoughtful comments and suggestions on an earlier draft of this paper.

References

1. Allen et al. (2003). Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, University of Massachusetts Amherst, September 2002. ACM SIGIR Forum 37(1), 31 – 47.
2. Belkin, N. J. (1980). Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science,* vol. 5: 133-143.

3. Bell, G. & Gemmell, J. (2003). Storage and media in the future when you store everything. Microsoft Research http://research.microsoft.com/~gbell/storeall.htm accessed on January 14, 2004.
4. Budzik, J., Hammond, K. & Birnbaum, L. (2001). Information access in context. *Knowledge Based Systems,* 14(1-2), 37-53.
5. Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., & Robbins, D. (2003). Stuff I've seen : A system for personal information retrieval and re-use. *Proceedings of SIGIR 2003 The 26th Annual International Conference on Retrieval and Development in Information Retrieval* (Toronto, Canada July 28-Aug 1, 2003). NY: ACM Press, 72-79.
6. Efron, M. Marchionini, G., & Zhang, J. (2003). Implications of the recursive representation problem for automatic concept identification in on-line governmental information. Proceedings of the ASIST Special Interest Group on Classification Research. (Los Angeles: Oct. 19-22, 2003).
7. Harter, S. (1992). Psychological relevance and information science. *Journal of the American Societyfor Information Science,* 43(9), 602-615.
8. Hearst, M. (1999). User interfaces and visualization. In R. Baeza-Yates & B. Ribeiro-Neto (Eds.) *Modern information retrieval.* pp. 257-323. NY : ACM Press and Addison-Wesley.
9. Heilprin, L. (1985). Paramorphism versus homomorphism in information science. In Laurence Heilprin (Ed), *Towardfoundations of information science.* White Plains, NY: Knowledge Industry Publications.
10. Ingwersen, P. (1992). Information retrieval interaction. London: Taylor Graham
11. Konstan, J. (2004). Introduction to recommender systems: Algorithms and evaluation. ACM Transactions on Information Systems, 22(1), 1-4.
12. Marchionini, G. An invitation to browse: Designing full-text systems for casual users. Canadian Journal of Information Science, 12(3/4), p. 69-79, 1987.
13. Meadow, C., Cerny, B., Borgman, C., & Case. D. (1989). Online access to knowledge: System design. *Journal of the American Societyfor Information Science,* 40(2), 86-98.
14. Mitra, M., & Singhal, A., & Buckley, C. (1998). Improving automatic query expansion. *Proceedings of SIGIR 1998 The 21st Annual International Conference on Retrieval and Development in Information Retrieval* (Melbourne, Australia Aug 24-28, 1998). NY: ACM Press, 206-214.
15. Robertson, S. (1997). Overview of the okapi projects. Journal of Documentation, 53(1), 3–7.
16. Salton, G. & Buckley, C. (1990). Improving retrieval performance by relevance feedback, *Journal of the American Society for Information Science,* 41(4), 288-297.
17. Saracevic, T. (1975). Relevance: A review of and a framework for thinking on the notion in information science, *Journal of the American Societyfor Information Science,* 26 321-343.
18. Schamber, L., Eisenberg, M. & Nilan, M. (1990). A re-examination of relevance: Toward a dynamic, situational definition, *Information Processing & Management,* 26(6), 755-776.
19. Shneiderman, B. (1994), Dynamic queries for visual information seeking, *IEEE Software* 11(6), 70-77.
20. Shneiderman, B., (1983). Direct manipulation: A step beyond programming languages, IEEE Computer 16(8), 57-69.

IR and AI: Traditions of Representation and Anti-representation in Information Processing

Yorick Wilks

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, S1 4DP.
Y.Wilks@dcs.shef.ac.uk

Abstract. The paper discusses the traditional, and ongoing, question as to whether natural language processing (NLP) techniques, or indeed and representational techniques at all, aid in the retrieval of information, as that task is traditionally understood. The discussion is partly a response to Karen Sparck Jones' (1999) claim that artificial intelligence, and by implication NLP, should learn from the methodology of Information Retrieval (IR), rather than vice versa, as the first sentence above implies. The issue has been made more interesting and complicated by the shift of interest from classic IR experiments with very long queries to Internet search queries which are typically of two highly ambiguous terms. This simple fact has changed the assumptions of the debate. Moreover, the return to statistical and empirical methods with NLP have made it less clear what an NLP technique, or even a "representational" method, is. The paper also notes the growth of "language models" within IR and the use of the term "translation" in recent years to describe a range of activities, including IR, and which constitutes rather the opposite of what Sparck Jones was calling for.

1 Introduction

Artificial Intelligence (AI), or at least non-Connectionist non-statistical AI, remains wedded to representations, their computational tractability and their explanatory power; and that normally means the representation of propositions in some more or less logical form. Classical Information Retrieval (IR), on the other hand, often characterised as a "bag of words" approach to text, consists of methods for locating document content independent of any particular explicit structure in the data. Mainstream IR is, if not dogmatically anti-representational (as are some statistical and neural net-related areas of AI and language processing), at least not committed to any notion of representation beyond what is given by a set of index terms, or strings of index terms along with figures themselves computed from text that may specify clusters, vectors or other derived structures. This intellectual divide over representations and their function goes back at least to the Chomsky versus Skinner debate, which was always presented by Chomsky in terms of representationalists versus barbarians, but was in fact about simple and numerically-based structures versus slightly more complex ones. Bizarre changes of allegiance took place during later struggles over the same issue, as when IBM created a machine translation (MT)

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 12–26, 2004.

system CANDIDE [8]. based purely on text statistics and without any linguistic representations, which caused those on the representational side of the divide to cheer for the old-fashioned symbolic MT system SYSTRAN in its DARPA sponsored contests with CANDIDE, although those same researchers had spent whole careers dismissing the primitive representations that SYSTRAN contained, nonetheless it was symbolic and representational and therefore on their side in this more fundamental debate! In those contests SYSTRAN always prevailed over CANDIDE for texts over which neither system had been trained, which may or may not have indirect implications for the issues under discussion here.

Winograd [57] is often credited in AI with the first natural language processing system (NLP) firmly grounded in representations of world knowledge yet, after his thesis, he effectively abandoned that assumption and embraced a form of Maturana's autopoesis doctrine [see 58], a biologically-based anti-representationalist position that holds, roughly, that evolved creatures cannot contain or manipulate representations. On such a view the Genetic Code is misnamed, which is a position with links back to the philosophy of Heidegger (whose philosophy Winograd began to teach at that period at Stanford in his NLP classes) as well as Wittgenstein's view that messages, representations and codes necessarily require intentionality, which is to say a sender, an insight that spawned the speech act movement in linguistics and NLP, and remains the basis of Searle's position that there cannot therefore be AI at all, as computers cannot have intentionality. The same insight is behind Dennett's more recent view that evolution undermines AI, as it does so much else. The debate within AI itself over representations, as within its philosophical and linguistic outstations, is complex and unresolved. The revived Connectionist/neural net movement of the 1980's brought some clarification of the issue within AI, partly because it came in both representationalist (localist) and non-representationalist (distributed) forms, which divided on precisely this issue. Matters were sometimes settled not by argument or experiment but by declarations of faith, as when Charniak said that whatever the successes of Connectionism he didn't like it because it didn't give him any perspicuous representations with which to understand the phenomena of which AI treats.

Within psychology, or rather computational psychology, there have been a number of recent assaults on the symbolic reasoning paradigm of AI-influenced Cognitive Science, including areas such as rule-driven expertise which was an area where AI, in the form of Expert Systems, was thought to have had some practical success. In an interesting revival of classic associationist methods, Schvaneveldt developed an associative network methodology for the representation of expertise (Pathfinder, 1990) producing a network whose content is extracted directly from subjects' responses and whose predictive powers in classic expert systems environments is therefore a direct challenge to propositional-AI notions of human expertise and reasoning. Within the main AI symbolic tradition, as I am defining it, it was simply inconceivable that a complex cognitive task, like controlling a fighter plane in real time, given input of a range of discrete sources of information from instruments, could be other than a matter for constraints and rules over coded expertise. There was no place there for a purely associative component based on numerical strengths of association or (importantly for his Pathfinder networks) on an overall statistical measure of clustering that establishes the Pathfinder network from the subject-derived data in the first place. Its challenge to traditional AI can be guaged from John

McCarthy's classic response to any attempt to introduce statistical notions into 1970's AI: "Where do all these numbers COME FROM?".

The Pathfinder example is highly relevant here, not only for its direct challenge to a core area of old AI, where it felt safe, as it were, but because the clustering behind Pathfinder networks was in fact very close, formally, to the clump theory behind the early IR work such as Sparck Jones [46] and others. Schvaneveldt [44] and his associates later applied the same Pathfinder networks to commercial IR after applying them to lexical resources like LDOCE. There is thus a direct algorithmic link here between the associative methodology in IR and its application in an area that took AI on directly in a core area. It is Schvaneveldt's results on knowledge elicitation by these methods from groups like pilots, and the practical difference such structure make in training, that constitute their threat to propositionality here. This is no unique example of course: even in older AI one thinks of Judea Pearl's long advocacy [38] of weighted networks to model beliefs, which captured (as did fuzzy logic and assorted forms of Connectionism since) the universal intuition that beliefs have strengths, and that these seem continuous in nature and not merely one of a set of discrete strengths, and that it is very difficult indeed to combine any system expressing this intuition with central AI notions of machine reasoning.

2 Background: Information Extraction (IE) as a Task and the Adaptivity Problem

In this paper, I am taking IE as a paradigm, naive though it still is, of an information processing technology separate from IR, formally separate, at least, in that one returns documents or document parts, and the other linguistic or data-base structures, although one must always bear in mind that virtually all IE search rests on a prior IR retrieval of relevant documents or paragraphs. IE is a technique which, although still dependent of superficial linguistic methods of text analysis, is beginning to incorporate more of the inventory of AI techniques, particularly knowledge representation and reasoning, as well as, at the same time, finding that some of its rule-driven successes can be matched by new machine learning techniques using only statistical methods (see below on named entities).

IE is an automatic method for locating facts for users in electronic documents (e.g. newspaper articles, news feeds, web pages, transcripts of broadcasts, etc.) and storing them in a data base for processing with techniques like data mining, or with off-the-shelf products like spreadsheets, summarisers and report generators. The historic application scenario for Information Extraction is a company that wants, say, the extraction of all ship sinkings, from public news wires in any language world-wide, and put into a single data base showing ship name, tonnage, date and place of loss etc. Lloyds of London had performed this particular task with human readers of the world's newspapers for a hundred years. The key notion in IE is that of a "template": a linguistic pattern, usually a set of attribute-value pairs, with the values being text strings. The templates are normally created manually by experts to capture the structure of the facts sought in a given domain, which IE systems then apply to text corpora with the aid of extraction rules that seek fillers in the corpus, given a set of syntactic, semantic and pragmatic constraints. IE has already reached the level of success at which Information Retrieval and Machine Translation (on differing measures, of course) have proved commercially viable. By general agreement, the main barrier to wider use and commercialization of IE is the relative inflexibility of its basic template concept: classic IE relies on the user having an already developed set of templates, as was the case with intelligence analysts in US Defence agencies from where the technology was largely developed. Yet this is not always the case in areas appropriate for the commercial deployment of IE. The intellectual and practical issue now is how to develop templates, their filler subparts (such as named entities or NEs), the rules for filling them, and associated knowledge structures, as rapidly as possible for new domains and genres. IE as a modern language processing technology was developed largely in the US. but with strong development centres elsewhere [13, 22, 25, 16]. Over 25 systems, world wide, have participated in the recent DARPA-sponsored MUC and TIPSTER IE competitions, most of which have a generic structure [25]. Previously unreliable tasks of identifying template fillers such as names, dates, organizations, countries, and currencies automatically – often referred to as TE, or Template Element, tasks have become extremely accurate (over 95% accuracy for the best systems). These core TE tasks have been carried out with very large numbers of hand-crafted linguistic rules.

Adaptivity in the MUC development context has meant beating the one-month period in which competing centres adapt their system to new training data sets provided by DARPA; this period therefore provides a benchmark for human-only adaptivity of IE systems. Automating this phase for new domains and genres, now constitutes the central problem for the extension and acceptability of IE in the commercial world and beyond the needs of the military sponsors who created it. The problem is of interest in the context of this paper, to do with the relationship of AI and IR techniques, because attempts to reduce the problem have almost all taken the form of introducing another area of AI techniques into IE, namely those of machine learning, and which are statistical in nature, like IR but unlike core AI. However, most of those used have been supervised techniques, which do tend to assume the need for some form of human-assignable representations.

3 Previous Work on ML and Adaptive Methods for IE

The application of Machine Learning methods to aid the IE task goes back to work on the learning of verb preferences in the Eighties by Grishman and Sterling [23] and Lehnert [30], as well as early work at MITRE on learning to find named expressions (NEs) [6]. The most interesting developments since then have been a series of extensions to the work of Lehnert and Riloff on Autoslog [40], the automatic induction of a lexicon for IE. This tradition of work goes back to an AI notion that might be described as lexical tuning, that of adapting a lexicon automatically to new senses in texts, a notion discussed in [56] and going back to work like Wilks [54] and Granger [19] on detecting new preferences of words in texts and interpreting novel lexical items from context and stored knowledge. This notion is important, not only for IE in general but in particular as it adapts to traditional AI tasks like Question Answering, now also coming within the IR remit (see below on TREC). There are also strong similarities between these forms of lexicon development and tuning done within AI/NLP and recent activities by e.g. Grefenstette and Hearst [21] on building massive IR lexicons from texts. The Autoslog lexicon development work is also described as a method of learning extraction rules from <document, filled template> pairs, that is to say the rules (and associated type constraints) that assign the fillers to template slots from text. These rules are then sufficient to fill further templates from new documents. No conventional learning algorithm was used by Riloff and Lehnert but, since then, Soderland has extended this work by using a form of Muggleton's ILP (Inductive Logic Programming) system for the task, and Cardie [10] has sought to extend it to areas like learning the determination of coreference links. Muggleton's [37] learning system at York has provided very good evaluated figures indeed (in world wide terms) in learning part of speech tagging and is being extended to grammar learning. Muggleton also has experimented with user interaction with a system that creates semantic networks of the articles and the relevant templates, although so far its published successes have been in areas like Part-of- Speech tagging that are not inherently structural (in the way template learning arguably is). Grishman at NYU [1] and Morgan [35] at Durham have done pioneering work using user interaction and definition to define usable templates, and Riloff [41] has attempted to use some version of user-feedback methods of Information Retrieval, including user-judgements of negative and positive <document, filled template> pairings.

3.1 Supervised Template Learning

Brill-style transformation-based learning methods are one of the few ML methods in NLP to have been applied above and beyond the part-of-speech tagging origins of virtually all ML in NLP. Brill's original application triggered only on POS tags; later [7] he added the possibility of lexical triggers. Since then the method has been extended successfully to e.g. speech act determination [9] and a Brill-style template learning application was designed by Vilain [51]. A fast implementation based on the compilation of Brill-style rules to deterministic automata was developed at Mitsubishi labs [42, 14]. The quality of the transformation rules learned depends on factors such as:

1. the accuracy and quantity of the training data;
2. the types of pattern available in the transformation rules;
3. the feature set available used in the pattern side of the transformation rules.

The accepted wisdom of the machine learning community is that it is very hard to predict which learning algorithm will produce optimal performance, so it is advisable to experiment with a range of algorithms running on real data. There have as yet been no systematic comparisons between these initial efforts and other conventional machine learning algorithms applied to learning extraction rules for IE data structures (e.g. example-based systems such as TiMBL [15] and ILP [36].

3.2 Unsupervised Template Learning

We should remember the possibility of unsupervised notion of template learning: in a Sheffield PhD thesis Collier [12] developed such a notion, one that can be thought of as yet another application of the old technique of Luhn [32] to locate statistically significant words in a corpus and use those to locate the sentences in which they occur as key sentences. This has been the basis of a range of summarisation algorithms and Collier proposed a form of it as a basis for unsupervised template induction, namely that those sentences, with corpus-significant verbs, would also contain sentences corresponding to templates, whether or not yet known as such to the user. Collier cannot be considered to have proved that such learning is effective, only that some prototype results can be obtained. This method is related, again via Luhn's original idea, to recent methods of text summarisation (e..g the British Telecom web summariser entered in DARPA summarisation competitions) which are based on locating and linking text sentences containing the most significant words in a text, a very different notion of summarisation from that discussed below, which is derived from a template rather than giving rise to it.

4 Linguistic Considerations in IR

Let us now quickly review the standard questions, some for over 30 years, in the debate about the relevance of symbolic or linguistic (or AI taken broadly) considerations in the task of information retrieval. Note immediately that this is not the reverse question touched on in the historical review above as to the relevance of IR-type methods or traditional NLP processing tasks, like machine translation and lexical structure, and which in its wider form concern the relevance of statistical methods to NLP in general. Note too that, even in the form in which we shall discuss it, the issue is not one between high-level AI and linguistic techniques on the one hand, and IR statistical methods on the other for, as the last section showed, the linguistic techniques normally used in areas like IE have in general been low-level, surface orientated, pattern matching techniques, as opposed to more traditional concerns of AI and linguistics with logical and semantic representations. So much has this been the case that linguists have in general taken no notice at all of IE, deeming it a set of heuristics almost beneath notice, and contrary to all long held principles about the necessity for general rules of wide coverage. Most IE has been a minute study of special cases and rules for particular words of a language, such as those involved in template elements (countries, dates, company names etc.). Again, since IE has also made extensive use of statistical methods, directly and as part of ML techniques, one cannot simply contrast statistical with linguistic methods in IE as Sparck Jones [47] does when discussing IR. In this connection, one must mention one of the most recent successes of purely statistical methods in IE, namely the BBN trained named entity finder, which is wholly statistical and producing results comparable with the large sets of grammatical rules just mentioned. That said, one should note that some IE systems that have performed well in MUC/TIPSTER-Sheffield's old LaSIE system would be an example [16] did also make use of complex domain ontologies, and general rule-based parsers. Yet, in the data-driven computational linguistics movement at the moment, one much wider than IE proper, there is a goal of seeing how far complex and "intensional" phenomena of semantics and pragmatics (e.g. dialogue pragmatics in [9]) can be treated by statistical methods. A key high-level IE task at the moment is co-reference, a topic that one might doubt could ever fully succumb to purely data-driven methods since the data is so sparse and the need for inference methods seems so clear. One can cite classic examples like:

{A Spanish priest} was charged here today with attempting to murder the Pope. {Juan Fernandez Krohn}, aged 32, was arrested after {a man armed with a bayonet} approached the Pope while he was saying prayers at Fatima on Wednesday night. According to the police, {Fernandez} told the investigators today that he trained for the past six months for the assault. He was alleged to have claimed the Pope "looked furious" on hearing {the priest's} criticism of his handling of the church's affairs. If found guilty, {the Spaniard} faces a prison sentence of 15-20 years.(The London Times 15 May 1982, example due to Sergei Nirenburg)

This passage contains six different phrases {enclosed in curly brackets} referring to the same person, as any reader can see, but which seem a priori to require knowledge and inference about the ways in which individuals can be described. There are three standard techniques in terms of which this possible infusion (of possible NLP techniques into IR) have been discussed, and I will then add a fourth.

i. Prior WSD (automatic word sense disambiguation) of documents by NLP techniques i.e. so that text words or some designated subset of them are tagged to particular senses in the form of a document to be retrieved by IR engines.

ii. The use of thesauri in IR and NLP as an intellectual and historical link between them.

iii. The prior analysis of queries and document indices so that their standard forms for retrieval reflect syntactic dependencies that could resolve classic ambiguities not of type (i) above.

Topic (i) is now mostly a diversion as regards our main focus of attention in this paper; even though large-scale WSD is now an established technology at the 95% accuracy level [44], there is no reason to believe it bears on the issue to hand, largely because the methods for document relevance used by classic IR are in fact very close to some of the algorithms used for WSD as a separate task (in e.g. Yarowsky [59, 60]). IR may well not need a WSD cycle because it has one as part of the retrieval process itself, certainly when using long queries as in TREC.

This issue has been clouded by the "one sense per discourse" claim of Yarowsky [59, 60], which has been hotly contested by Krovetz [28] who has had had no difficulty showing that Yarowsky's figures (that a very high percentage of words occur in only one sense in any document) are just wrong and that, outside Yarowsky's chosen world of encyclopedia articles, is not at all uncommon for words to appear in the same document bearing more than one sense on different occasions of use.

Note that this dispute is not one about symbolic versus statistical methods for task, let alone AI versus IR. It is about a prior question as to whether there is any serious issue of sense ambiguity in texts to be solved at all, and by any method. In what follows I shall assume Krovetz has the best of this argument and that the WSD problem, when it is present, cannot be solved, as Yarowsky claimed in the one-sense-per-discourse paper, by assuming that only one act of sense resolution was necessary per text. Yarowsky's claim, if true, would make it far more plausible that IR distributional methods were adequate for resolving the sense of component words in the act of retrieving documents, because sense ambiguity resolution is at the document level, as Yarowsky's claim makes clear.

If Krovetz is right then sense ambiguity resolution is still a local matter within a document and one cannot have confidence that any word is univocal within a document, nor that a document-span process will resolve such ambiguity and hence one will have less confidence that standard IR processes resolve such terms if they are crucial to the retrieval of a document. One will expect, a priori, that this will be one cause of lower precision in retrieval, and the performance of web engines confirms this anecdotally in the absence of any experiments going beyond Krovetz's own.

Let us now turn to (ii), the issue of thesauri: there is less in this link in modern times, although early work in both NLP and IR made use of a priori hand-crafted thesauri like Roget. Though there is still distinguished work in IR using thesauri in specialised domains, beyond their established use as user-browsing tools (e.g. [11]), IR moved long ago towards augmenting retrieval with specialist, domain-dependent and empirically constructed thesauri, while Salton [43] early on (1972) claimed that results with and without thesauri were much the same.

NLP has rediscovered thesauri at intervals, most recently with the empirical work on word-sense disambiguation referred to above, but has remained wedded to either Roget or more recent hand- crafted objects like WordNet [34]. The objects that go under the term thesaurus in IR and AI/NLP are now rather different kinds of thing, although in work like Hearst and Grefenstette (1992) an established thesaurus like WordNet has been used to expand a massive lexicon for IR, again using techniques not very different from the NLP work in expanding IE lexicons referred to earlier.

Turning now to (iii), the use of syntactic dependencies in documents, their indices and queries, we enter and large and vexed area, in which a great deal of IR work has been done within IR [45]. There is no doubt that some web search engines routinely make use of such dependencies: take a case like measurements of models as opposed to models of measurement where these might be taken to access different literatures, although the purely lexical content, or retrieval based only on single terms, might be expected to be the same. In fact they get 363 and 326 hits respectively in Netscape but the first 20 items have no common members. One might say that this case is of type (i), i.e. WSD, since the difference between them could be captured by, say, sense tagging "models" by the methods of (i), whereas in the difference between the influence of X on Y and (for given X and Y) The influence of Y on X one could not expect WSD to capture the difference, if any, if X and Y were 'climate' and 'evolution' respectively, even though these would then be quite different requests.

These are standard types of example and have been the focus of attention both of those who believe in the role of NLP techniques in the service of IR (e.g. Strzalkowski and Vauthey,[50]), as well as those like Sparck Jones [47] who do not accept that such syntactically motivated indexing has given any concrete benefits not available by other, non-linguistic, means. Sparck Jones' paper is a contrast between what she call LMI (Linguistically Motivated Information Retrieval) and NLI (Non-Linguistically etc.), where the former covers the sorts of efforts described in this paper and the latter more 'standard' IR approaches. In effect, this difference always comes down to one of dependencies within, for example, a noun phrase marked either explicitly by syntax or by word distance windows. So for example, to use her own principal example:

URBAN CENTRE REDEVELOPMENTS could be structured (LMI-wise) as REDEVELOPMENTS of [CENTRE of the sort URBAN] or as a search for a window in full text as (NLI-wise) [URBAN =0 CENTRE]<4 REDEVELOPMENTS where the numbers refer to words that can intrude in a successful match.

The LMI structure would presumably be imposed on a query by a parser, and therefore only implicitly by a user, while the NLI window constraints would again presumably be imposed explicitly by the user. It is clear that current web engines use both these methods, with some of those using LMI methods derived them directly from DARPA-funded IE/IR work (e.g. NetOWL and TextWise). The job advertisements on the Google site show clearly that the basic division of methods at the basis of this paper have little meaning for the company, which sees itself as a major consumer of LMI/NLP methods in improving its search capacities.

Sparck Jones' conclusion is one of measured agnosticism about the core question of the need for NLP in IR: she cites cases where modest achievements have been found, and others where LMI systems' results are the same over similar terrain as NLI ones. She gives two grounds for hope to the LMIers: first, that most such results are over queries matched to abstracts, and one might argue that NLP/LMI would come into play more with access to full texts, where context effects might be on a greater scale and, secondly, that some of the more negative results may have been because of the long queries supplied in TREC competitions, and that shorter more realistic and user-derived, queries might show a greater need for NLP. The development of Google, although proprietary, allows one to guess that this has in fact been the case in the world of Internet searches. On the other hand, she offers a general remark (and I paraphrase substantially here) that IR is after all a fairly coarse task and it may be one not in principle optimisable by any techniques beyond certain limits, perhaps those we have already; the suggestion being that other, possibly more sophisticated, techniques should seek other information access tasks and leave IR as it is. This demarcation has distant analogies to one made within the word-sense discrimination research mentioned earlier, namely that it may not be possible to push figures much above where they now are, and therefore not possible to discriminate down to the word sense level, as oppose to the cruder homograph level, where current techniques work best, on the ground that anything "finer" is a quite different kind of job, and not a purely linguistic or statistical one, but rather one for future AI, if anything. Sparck Jones [48] developed these views further, and as far as to call on AI in general to adopt more IR-like methodologies. In so far as that means evaluation techniques, no one could possibly disagree but, curiously, in the area under discussion one might even say the battle has gone the other way. Since about the time of her own paper, have come a stream of papers, starting with Berger and Lafferty [4] with titles like "Information Retrieval as Statistical Translation" in which, in a curious and inverted sense, machine translation is being taken as a desirable model of IR to conform to, which is certain the reverse of the shift Sparck Jones wanted.

As always, things are not as radical as they seem: the genesis of the Berger and Lafferty work, under the broad heading of "language models in IR", was the IBM statistical translation work at IBM under Jelinek referred to earlier, and the word "translation" in the Berger/Lafferty title refers in a sense to any technique which considers two strings in relationship to each other as, indifferently, translations of each other or retrievals of each other. This is undoubtedly a touch of sleight of hand here, since translation is normally considered a symmetrical relationship, but retrieval is not, since documents are not normally considered as retrieving queries. However, and quibbles aside, this approach has now even suggested considering question-answering as a form of translation (as it been seen as a form of IR for quite a while) and in that case the asymmetry is not so striking (see [5]). All this work remains statistical, as indeed is so much of NLP and AI these days, but there is clearly an element here of NLP techniques, however construed, being applied to IR (rather than the reverse) even if much of this is achieved by a widening or redefinition of the core IR task itself. It is worth noting in that connection that the development of IR as cross-language task has also, and inevitably, increased the role of NLP techniques, and made access to NLP resources, such as lexicons seems natural and obvious, as is shown in work like Gollins and Sanderson's [18] cross-language retrieval by what they call "language triangulation."

(iv) The use of proposition-like objects as part of document indexing is a notion which, if sense can be given to it, would be a major revival of NLP techniques in aid of IR. It is an extension of the notion of (iii) above, which could be seen as an attempt to index documents by template relations, e.g. if one extracts and fills binary relation templates (X manufactures Y; X employs Y; X is located in Y) so that documents could be indexed by these facts in the hope that much more interesting searches could in principle be conducted (e.g. find all documents which talk about any company which manufactures drug X, where this would be a much more restricted set than all those which mention drug X). One might then go on to ask whether documents could profitably be indexed by whole scenario templates in some interlingual predicate form (for matching against parsed queries) or even by some chain of such templates, of the kind extracted as a document summary by co-reference techniques (e.g.[2]).

Few notions are new, and ideas of applying semantic analysis to IR in some manner, so as to provide a complex structured (even propositional) index, go back to the earliest days of IR. In the 1960s researchers like Gardin [17], Gross [24] and Hutchins [26] developed complex structures derived from MT, from logic or "text grammar" to aid the process of providing complex contentful indexes for documents, of the order of magnitude of modern IE templates. Of course, there was no hardware or software to perform searches based on them, though the notion of what we would now call a full text search by such patterns so as to retrieve them go back at least to Wilks [52, 53] even though no real experiments could be carried out at that time. Gardin's ideas were not implemented in any form until (Bely et al, [3]) which was also inconclusive. Mauldin [33], within IR, implemented document search based on case-frame structures applied to queries (ones which cannot be formally distinguished from IE templates), and the indexing of texts by full, or scenario, templates appears in Pietrosanti and Graziadio [39]. The notion is surely a tempting one, and a natural extension of seeing templates as possible content summaries of the key idea in a text [2]. If a scenario template or a chain of them, can be considered as a summary then it could equally well, one might think, be a candidate as a document index. The problem will be, of course, as in the summarisation work by such methods, what would cause one to believe that an a priori template would or could capture they key item of information in a document, at least without some separate and very convincing elicitation process that ensured that the template corresponded to some class of user needs, but this is an empirical question and one being separately evaluated by summarisation competitions.

Although this indexing-by-template idea is in some ways an old one, it has not been aired lately, and like so much in this area, has not been conclusively confirmed or refuted as an aid to retrieval. It may be time to revive it again with the aid of new hardware, architectures and techniques–after all, connectionism/neural nets was only an old idea revived with a new technical twist, and it has had a ten year or more run in its latest revival. What seems clear at the moment is that, in the web and Metadata world, there is an urge to revive something along the lines of “get me what I mean, not what I say” (see Jeffrey, [27]). Long-serving IR practitioners will wince at this, but to many it must seem worth a try, since IE does have some measurable and exploitable successes to its name (especially Named Entity finding) and, so the bad syllogism goes, Metadata is data and IE produces data about texts, so IE can produce Metadata.

5 Question Answering within TREC

No matter what the limitation on crucial experiments so far, another place to look for evidence of the current of NLP/AI influence on IR might be the new Question-Answering track within TREC 1999, already touched on above in connection with IRs influence on AI/NLP, or vice versa. Question answering is one of the oldest and most traditional AI/NLP tasks (e.g.[20, 29]) but can hardly be considered solved by those structural methods. The conflation, or confusion, of the rival methodologies distinguished in this paper, can be clearly seen in the admitted possibility, in the TREC QA competition, of providing ranked answers, which fits precisely with the continuous notion of relevance coming from IR , but is quite counterintuitive to anyone taking a common sense view of questions and answers, on which that is impossible. It is a question master who provides a range of differently ranked answers on the classic QA TV shows, and the contestant who must make a unique choice (as opposed to re-presenting the proffered set!). That is what answering a question means; it does not mean “the height of St Pauls is one of [12, 300, 365, 508]feet”! A typical TREC question was “Who composed Eugene Onegin?” and the expected answer was Tchiakowsky which is not a ranking matter, and Gorbachev, Glazunov etc. are no help. There were examples in this competition that brought out the methodological difference between AI/NLP one the one hand, and IR on the other, with crystal clarity: answers could be up to 250 bytes long, so if your text-derived answer was A, but wanting to submit 250 bytes of answer meant that you, inadvertently, could lengthen that answer rightwards in the text to include the form (A AND B), in which case your answer would become wrong in the very act of conforming to format. The anecdote is real, but nothing could better capture the absolute difference in the basic ontology of the approaches: one could say that AI, Linguistics and IR were respectively seeking propositions, sentences and byte-strings and there is no clear commensurability between the criteria for determining the three kinds of entities. For this first TREC question answering competition, comparative results across sites will undoubtedly be forthcoming soon though this will be taken as only bench marking for subsequent years, and not the settling of this deep ideological divide, one which web entrepreneurs cheerfully ignore, taking their techniques from wherever they can. But I suggest, for anyone interested in the issue, this TREC track is the one to watch. There is an interesting parallel, by the way, for this byte-approach to QA: in 1996 CMU entered into the Loebner Turing competition, a computer discussant that answered questions, and produced all responses, by a closest match word-window algorithm into a large newspaper corpus. It didn't do very well, but not nearly as disastrously as those in the LMI school would have predicted and wanted! One should remember too that very early attempts were made, within the IR tradition, (by O'Connor and Miller) to use retrieval of micro-texts as a form of QA.

6 Conclusion

One can make quite definite conclusions but no predictions, other than those based on hope. Of course, after 40 years, IR ought to have improved more than it has its overall Precision/Recall figures are not very different from decades ago. Yet, as Sparck Jones has shown, there is no clear evidence that NLP adds more than marginal improvements to IR, which may be a permanent condition, or one that will change with full text search, and a different kind of user- derived query, and Google may be one place to watch for this technology to improve strongly. It may also be worth someone in the IE/LMI tradition trying out indexing-by-scenario templates for IR, since it is, in one form or another, an idea that goes back to the earliest days of IR and NLP, but remains untested. It is important to remember as well, that there is a deep cultural division in that AI remains, in part at least, agenda driven: in that certain methods are to be shown effective. IR, like all statistical methods in NLP as well, remains more result-driven, and the clearest proof of this is that (with the honourable exception of machine translation) all evaluation regimes have been introduced in connection with statistical methods, often over strong AI/linguistics resistance. In IE proper, one can be moderately optimistic that fuller AI techniques of ontology, knowledge representation and inference, will come to play a stronger role as the basic pattern matching and template element finding is subject to efficient machine learning. One may be moderately optimistic, too, that IE may be the technology vehicle with which old AI goals of adaptive, tuned, lexicons and knowledge bases can be pursued, IE may also be the only technique that will ever provide a substantial and consistent knowledge base from texts, as CYC [31] has failed to do over twenty years. The traditional AI/QA task, now brought within TREC, may yield to IR, IE methods or some mixture of the two, but it will be a fascinating struggle. The curious tale above, of the use of translation with IR and QA work suggests that matters are very flexible at the moment and it may not be possible to continue to draw the traditional demarcations between these close and merging NLP applications IE, MT, QA and so on.

Acknowledgements. The paper has benefited from discussions with Mark Sanderson, Rob Gaizauskas, Ted Dunning and others, but the errors are all mine of course.

References

Agichtein E., Grishman R., Borthwick A. and Sterling J. 1998. Description of the named entity system as used in MUC-7. In Proceedings of the MUC-7 Conference, NYU.

Azzam, S. Humphreys, K. and Gaizauskas, R. 1999. using coreference chains for text summarization. Proc. ACL Workshop on Coreference and its Applications, Maryland.

Bely, N. Borillo, A, Virbel, J and Siot-Decauville, N. 1970. Procedures d'analyse semantique appliquees a la documentation scientifique. Gauthier-Villars: Paris.

Berger, A., Lafferty, J. 1999. Information retrieval as statistical translation. SIGIR 1999.

Berger, A. et al. 2000. Bridging the lexical chasm: statistical approaches to question answering. SIGIR 2000.

Bikel D., Miller S., Schwartz R., and Weischedel R. 1997. Nymble: a High-Performance Learning Name-finder. In Proceedings of the Fifth conference on Applied Natural Language Processing.

Brill E. 1994. Some Advances in Transformation-Based Part of Speech Tagging. In Proceedings of the Twelfth National Conference on AI (AAAI-94), Seattle, Washington.

Brown, P. F. and Cocke, John, 1989. A Statistical Approach to Machine Translation, IBM Research Division, T.J. Watson Research Center, RC 14773.

Carberry S., Samuel K. and Vijay-Shanker K. 1998. Dialogue act tagging with transformation-based learning. In Proceedings of the COLING-ACL 1998 Conference, volume 2, pages 1150–1156, Montreal, Canada, 1998.

Cardie C. 1997. Empirical methods in information extraction. AI Magazine, 18(4),Special Issue on Empirical Natural Language Processing.

Chiaramella, Y., Nie, J. 1990. A retrieval model based on an extended modal logic and its application to the RIME experimental approach, in Proceedings of the 13th ACM International Conference on Research and Development in Information Retrieval (SIGIR): 25-43

Collier R. 1998. Automatic Template Creation for Information Extraction. PhD thesis, University of Sheffield Computer Science Dept., UK.

Cowie J., Guthrie L., Jin W., Odgen W., Pustejowsky J., Wanf R., Wakao T., Waterman S. and Wilks Y. 1993. CRL/Brandeis: The Diderot System. In Proceedings of Tipster Text Program (Phase I). Morgan Kaufmann.

Cunningham H. 1999. JAPE – a Java Annotation Patterns Engine. Technical Report, Department of Computer Science, University of Sheffield.

Daelemans W., Zavrel J., van der Sloot K., and van den Bosch A. 1998. TiMBL: Tilburg memory based learner version 1.0. Technical report, ILK Technical Report 98-03.

Gaizauskas, R. and Wilks, Y. 1997. Information Extraction: beyond document retrieval. Journal of Documentation.

Gardin, J. 1965. Syntol. New Brunswick, NJ: Rutgers Graduate School of Library Science.

Gollins, T., Sanderson, M. 2001. Improving Cross Language Information Retrieval with triangulated translation. SIGIR 2001.

Granger, R. (1977) FOULUP: a program that figures out meanings of words from context. Proc. Fifth Joint Internat. Conf. on AI.

Green, B., Wolf. A., Chomsky, C., and Laughery, K. 1961. BASEBALL, an automatic question answerer. Proc. Western Joint Computer Conference 19, 219-224

Grefenstette, G., Hearst, M.A 1992. Method for Refining Automatically-Discovered Lexical Relations: Combining Weak Techniques for Stronger Results. In Weir (ed.) Statistically-based natural language programming techniques, Proc. AAAI Workshop, AAAI Press, Menlo Park, CA.

Grishman R. 1997. Information extraction: Techniques and challenges. In M-T. Pazienza, editor, Proceedings of the Summer School on Information Extraction (SCIE-97), LNCS/LNAI. Springer-Verlag.

Grishman R. and Sterling J. 1992. Generalizing automatically generated patterns. In Proceedings of COLING-92.

Gross, M. 1964. On the equivalence of models of language used in the fields of mechanical translation and information retrieval. Information Storage and Retrieval. 2(1).

Hobbs J.R. 1993. The generic information extraction system. In Proceedings of the Fifth Message Understanding Conference (MUC-5), pages 87–91. Morgan Kaufman.

Hutchins, W. J. 1970 Linguistic processes in the indexing and retrieval of documents. Linguistics, 61.

Jeffrey, K. 1999. What's next in databases? ERCIM News (www.ercim.org) 39.

Krovetz, R. 1998 More than one sense per discourse. NEC Princeton NJ Labs., Research Memorandum.

Lehnert, W. 1977. A Conceptual Theory of Question Answering. Proc. Fifth IJCAI, Cambridge, MA. Los Altos: Kaufmann, 158-164.

Lehnert W., Cardie C., Fisher D., McCarthy J., and Riloff E. 1992. University of Massachusetts: Description of the CIRCUS system as used for MUC-4. In Proceedings of the Fourth Message Understanding Conference MUC-4, pages 282–288. Morgan Kaufmann.

Lenat, D., M. Prakash, and M. Shepherd. 1986. CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks, The AI Magazine, 6(4).

Luhn, H.P. 1957. A statistical approach to mechanized encoding and searching of literary information. IBM Journal of Research and Development, 1:309–317.

Mauldin, M. 1991. Retrieval performance in FERRET: a conceptual information retrieval system. SIGIR91

Miller, G. A. (ed.) 1990.WordNet: An on-line Lexical Database, In International Journal of Lexicography, 3(4).

Morgan R., Garigliano R., Callaghan P., Poria S., Smith M., Urbanowicz A., Collingham R., Costantino M., and Cooper C. 1995. Description of the LOLITA System as used for MUC-6. In Proceedings of the Sixth Message Understanding Conference (MUC-6), pages 71–86, San Francisco, Morgan Kaufmann.

Muggleton S. 1994. Recent advances in inductive logic programming. In Proc. 7th Ann. ACM Workshop on Comput. Learning Theory,Pages 3–11. ACM Press, New York, NY.

Muggleton S., Cussens J., Page D., and Srinivasan A. 1997. Using inductive logic programming for natural language processing. In Proceedings of in ECML, pages 25–34, Prague. Springer-Verlag. Workshop Notes on Empirical Learning of Natural Language Tasks.

Pearl, J. 1985. Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning, In Proceedings of the Cognitive Science Society (CSS-7).

Pietrosanti, E., and Graziadio, B. 1997. Extracting Information for Business Needs. Unicom Seminar on Information Extraction, London, March.

Riloff E. and Lehnert W. 1993. Automated dictionary construction for information extraction from text. In Proceedings of Ninth IEEE Conference on Artificial Intelligence for Applications, pages 93-99.

Riloff E. and Shoen J. 1995. Automatically acquiring conceptual patterns without an annotated corpus. In Proceedings of the Third Workshop on Very Large Corpora.

Roche E. and Schabes Y. 1995. Deterministic Part-of-Speech Tagging with Finite-State Transducers. Computational Linguistics, 21(2):227-254.

Salton, G. 1972 A new comparison between conventional indexing (MEDLARS) and automatic text processing (SMART). Journal of the American Society of Information Science, 23(2).

Schvaneveldt, R. (ed.) 1990. Pathfinder Networks: Theory and Applications. Ablex, Norwood, NJ.

Smeaton, A. and van Rijsbergen, C. 1988. Experiments in incorporating syntactic processing of user queries into a document retrieval strategy. Proc. 11th. ACM SIGIR.

Sparck Jones, K. 1966/1986. Synonymy and Semantic Classification. Edinburgh University Press, Edinburgh.

Sparck Jones, K. 1999a. What is the role of NLP in text retrieval. In Strzalkowski (ed.) Natural language Information Retrieval. Kluwer: New York.

Sparck Jones, K. 1999b. Information Retrieval and Artificial Intelligence. Artificial Intelligence Journal, vol. 114.

Stevenson, M. and Wilks, Y. 1999. Combining Weak Knowledge Sources for Sense Disambiguation Proceedings of the International Joint Conference for Artifical Intelligence (IJCAI-99)

Strzalkowski, T. and B. Vauthey, 1991. Natural Language Processing in Automated Information Retrieval, PROTEUS Project Memorandum. Department of Computer Science, New York University.

Vilain M. 1993. Validation of terminological inference in an information extraction task. In Proceedings of the 1993 ARPA Human Language Workshop.

Wilks, Y. 1964. Text Searching with Templates. Cambridge Language Research Unit Memo, ML. 156.

Wilks, Y. 1965. The application of CLRU's method of semantic analysis to information retrieval. Cambridge Language Research Unit Memo, ML. 173.

Wilks, Y. 1979. Frames, semantics and novelty. In Metzing (ed.) Frame Conceptions and Text Understanding. Berlin: de Gruyter.

Wilks, Y., 1998, Senses and Texts, In N. Ide (ed.) special issue of Computers and the Humanities.

Wilks, Y. and Catizone, R. 1999. Making information extraction more adaptive. In M-T. Pazienza (ed.) Proc. Information Extraction Workshop, Frascati.

Winograd, T. 1971. Understanding Natural language.

Winograd, T. and Flores, A. 1986. Understanding Computers and Cognition: A New Foundation for Design, Ablex: Norwood, NJ.

Yarowsky, D. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In Proc. COLING92, Nantes, France.

Yarowsky, D. 1995. Unsupervised word-sense disambiguation rivalling supervised methods, Proc. ACL-95.

A User-Centered Approach to Evaluating Topic Models

Diane Kelly[1], Fernando Diaz[2], Nicholas J. Belkin[3], and James Allan[2]

[1] SILS, University of North Carolina
Chapel Hill, NC, USA 27599
dianek@ils.unc.edu
[2] CIIR, University of Massachusetts, Amherst
Amherst, MA, USA 01003
{fdiaz,allan}@cs.umass.edu
[3] SCILS, Rutgers University
New Brunswick, NJ, USA 08901
nick@belkin.rutgers.edu

Abstract. This paper evaluates the automatic creation of personal topic models using two language model-based clustering techniques. The results of these methods are compared with user-defined topic classes of web pages from personal web browsing histories from a 5-week period. The histories and topics were gathered during a naturalistic case study of the online information search and use behavior of two users. This paper further investigates the effectiveness of using display time and retention behaviors as implicit evidence for weighting documents during topic model creation. Results show that agglomerative techniques - specifically, average-link clustering - provide the most effective methodology for building topic models while ignoring topic evidence and implicit evidence.

1 Introduction

A general problem for current interactive information retrieval (IR) systems is disambiguating the topic of interest to a searcher, given a statement of the person's information problem, typically posed as a rather brief query. One possible approach to this issue is to take advantage of the person's previous information seeking behaviors in order to identify topics which have been of interest to that person in the past. This could be done, for instance, by recording the documents (e.g. Web pages) that the person has looked at as a result of searching for information, and automatically classifying those pages according to topic models, derived from the language of the documents. A new search by the person could be associated with one or a few of such models, thereby effectively disambiguating the search topic, and providing a basis for searching for new documents which might be generated by the topic model(s). Language modeling and clustering techniques have proven useful for generating topic models in other domains [1,18]. However, the effectiveness of such techniques on personal collections has yet to be tested.

Another method of topic identification is to observe such behaviors as display (or dwell) time, or bookmarking, printing or otherwise saving or using documents. Given such evidence from previous and current behaviors, documents of current interest could be related to documents of past interest, and therefore to topic models.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 27–41, 2004.

The purpose of this paper is to explore a novel method of evaluating the accuracy of topic models which have been created using traditional language modeling and clustering approaches, and behavioral evidence, such as display time and retention. This method consists of comparing topic models created using these approaches with those created by users during a naturalistic study using self-identified topics.

2 Related Literature

Recently, language modeling has received much attention in the IR community [5]. In this framework, a collection of text data—documents or sets of documents—is considered to be sampled from an underlying generative process. Namely, we assume that the words in a document were all generated according to some topic model. A topic model is a probability distribution over words. For example, the topic "Iraq War" may have some probability of generating the terms "Bush", "Hussein", and "Iraq" and much lower probability of generating "cattle", "dance", and "salsa".

Figure 1 provides a graphical interpretation of this approach. A topic language model can be described using the urn metaphor. Each topic is considered an infinite collection of words which follow some distribution. A document is produced by iteratively picking words from this urn. Unfortunately, we do not have access to the true distribution of terms in this topic urn, but known, on-topic documents can be used as evidence for estimating this distribution. Fortunately, casting the task as model estimation allows one to use formalisms from statistics. Previous techniques such as the vector space model often relied a great deal on heuristics and hand-tuned parameters.

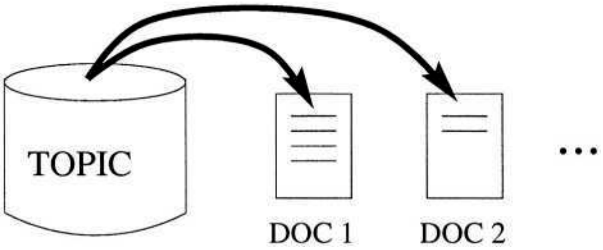


Fig. 1. Topic Language Model: A topic can be interpreted as an infinitely large collection of words. Documents are generated by choosing words from this urn.

The most relevant work for topic modeling has been conducted in the context of Topic Detection and Tracking [1]. This literature deals with the problem of tracking several topics in a stream of news articles. The community has produced several techniques for automatically clustering and detecting links between documents in a stream. Although more sophisticated language modeling systems have been developed for these tasks, the most successful approaches use straight-forward vector-space techniques [2]. Nevertheless, language modeling systems provide a formal methodology for estimating the topic models.

Several techniques have also attempted to create user-topic clusters in an IR setting. In these cases, clusters are constructed by incorporating explicit user feedback usually in an interactive IR setting [3,6,8]. In an environment of passive feedback, many authors have described the incorporation of disambiguating terminology from a user's search history [7,15]. These systems often build a user model and leverage this information to expand the query.

Other approaches to user modeling for personalized IR have used the user's online behaviors as implicit indicators of interest [4,10,12,14,16]. Typical approaches have used display time, retention (e.g. printing, saving, and bookmarking), scrolling and selection to identify relevant documents for feedback during a single search session. Behavioral evidence has also been used to cluster search results [9], but it has not been used as evidence for topic model construction.

The evaluation of topic models has typically consisted of various cluster or link detection measures [1]. These approaches often use annotator consensus as a baseline for evaluation. It is often difficult to assess how a technique will actually perform in "real-life" because of a lack of a user-centered evaluation metric. Instead, assumptions must be made about how a user would classify, label and evaluate documents. In the paper, we evaluate the accuracy of two language model-based clustering techniques with user-defined topic classes of web pages from personal web browsing histories. We further investigate the effectiveness of including behavioral evidence in the construction of these models.

3 Monitoring Study

The data used for the study reported in this paper was collected during a naturalistic case study of the online informaiton-seeking behaviors of two users during a five-week period. Users were provided with laptop computers and their activities were monitored with logging and evaluation software and online questionnaires.

We chose a naturalistic approach because we were interested in providing users with an opportunity to engage in multiple information seeking episodes over time, with tasks and topics that were germane to their personal interests, in familiar searching environments. The naturalistic approach also provided an advantage over web server log analysis because the identity of users could be maintained and various measurements could be collected during the observational period. Furthermore, information about intentions and specific tasks and topics could be gathered and associated with behaviors and documents. We chose to conduct two descriptive case studies because we were interested in gathering a large, detailed quantity of data. We do not claim our two users to be a sample, nor do we claim that our results generalize reliably to a larger population of users.

3.1 Users

Two volunteer users completed the five-week study. Both users were graduate students in a Master's of Library and Information Science program and held Bachelor's and Master's degrees in the Humanities. Both users had a high degree of self-assessed computer and online searching experience.

3.2 Instruments and Procedures

Each user was provided with an IBM ThinkPad equipped with the Windows 2000 operating system and standard utilities to use for the duration of the study. The laptops were equipped with client-side logging software that monitored and recorded users' interactions with the operating system and all other applications. The monitoring software was launched automatically each time the laptop was started, executed in stealth mode while the laptop was in operation and recorded information such as applications used, URLs visited, start, finish and elapsed times for interactions and all keystrokes. A proxy server captured all pages the user viewed while connected to the Internet.

Two types of behaviors were of interest to this study: display time and retention. In this study, display time was the length of time that a document was displayed in the user's active browser window. Display time was collected from the client-side logger, which indicated elapsed times for displaying a particular document. Since the client-side logger recorded active window data and all programs with which the user was interacting, we feel somewhat confident that this measure was accurate. While we cannot insure that the user was viewing the document and not attending to other offline activities, we are confident that display time data collected from a client-side logger is more reliable than that collected from a proxy server. Retention behaviors [13] included saving, printing, emailing or bookmarking, and were gathered directly from the client-side logger.

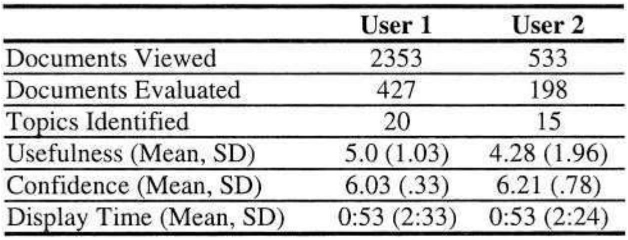
At the beginning of the study, users read and signed a consent form, which outlined the protocol of the study and informed users that all of their activities with the laptops would be monitored. An Entry Questionnaire, which elicited background information from the user, such as education and search experience, was administered. A Task and Topic Questionnaire was also administered that elicited the tasks and topics the user would be engaged with during the period of the study. Users were asked to think about their online activities in terms of tasks and topics. For example, a task might be shopping and the topic of this task might be clothing, or guitars. Another example task might be writing a research paper; the topic of this task might be political ecology and West Africa.

All pages that the user viewed while searching were captured by a proxy server. A content-based classification of page types was created based on a manual examination of 2000 pages to identify and select systematically the pages that were to be evaluated in the study. The goal was to eliminate pages such as ads, search pages, email pages, etc. Two independent coders validated this classification.

A Task and Topic Update Questionnaire was administered online each week of the study, which presented users with their previously identified tasks and topics and asked them to update the list through additions and/or deletions.

At the mid- and end-points of the study, the pages viewed up to that time were presented online to the users for evaluation. The instrument used for this evaluation displayed the text of one page at a time, a console which had two drop-down lists containing the user's tasks and topics and text boxes in the event that new tasks or topics needed to be added during the evaluation. Users were asked to classify each page that they viewed according to its task and topic and to evaluate the usefulness of the page using seven-point usefulness (1=not useful, 7=useful) and confidence scales (1=low, 7=high).

Table 1. Description of behavior and page evaluations



	User 1	User 2
Documents Viewed	2353	533
Documents Evaluated	427	198
Topics Identified	20	15
Usefulness (Mean, SD)	5.0 (1.03)	4.28 (1.96)
Confidence (Mean, SD)	6.03 (.33)	6.21 (.78)
Display Time (Mean, SD)	0:53 (2:33)	0:53 (2:24)

3.3 Results of the Monitoring Study

There were several types of data that we were interested in using from the monitoring study. We were interested in the users' self-identified topics and their classification of the documents that they viewed into each of these groups. We were interested in users' display time and retention behaviors. Finally, we were interested in the usefulness ratings that users associated with each page. In sum, the data from the monitoring study provided us with sets of documents that had been clustered into self-identified topics by users, and usefulness scores and behaviors for each document. In this study, we did not consider the task classes created by our users. While this information may be helpful in distinguishing topic classes, we leave it for future analysis.

A total of 2353 items were logged by the proxy for User 1 and 533 were logged for User 2. After screening the documents according to the classification described above, 427 (18%) were identified for evaluation by User 1 and 198 (36%) for User 2. Table 1 displays an overall description of the number of documents viewed and evaluated by each user, the number of topics identified, the mean usefulness and confidence of the pages evaluated and the mean display times. Interestingly, the mean display time for all documents for both users was identical. In general, both users were very confident with their evaluations of the documents that they viewed.

Users identified a range of topics with which they were engaged throughout the study. A list of topics for each user is displayed in Table 2. This table includes the number of documents viewed for each topic (D), the mean display time of documents for each topic (RT), the number of retention behaviors for each topic (RET) and the mean usefulness (Use). Many of these topics were related to libraries, since both users were Masters students in library and information science, and both users worked in libraries. Also, both users were concurrently enrolled in the same course, and several topics were related to the same course project. For example, both users identified the topic of "evaluation criteria," which was related to a course assignment about developing evaluation criteria to assess the usability of web resources. "Review material" and "book review research" also represented a particular course project that required students to identify and evaluate online sources for book reviews. Other topics represented users' specific content-based interests in libraries; User 1 studied theology-based texts, while User 2 studied classic texts and various materials associated with classic texts such as papyri. Other topics represented individual interests unrelated to the course, such as "eyeglasses," "sailing," and "recipe search."

We used all user-defined topic clusters in the present study as a baseline with which to evaluate clusters created using automatic techniques.

Table 2. Topics identified by User 1 and User 2 and characteristics of each

User 1					User 2				
Topic	D	RT	RET	Use	Topic	D	RT	RET	Use
Theology	55	00:38	1	4.45	General Interest	57	01:12	4	5.52
Perennials	11	01:00	0	5.00	College Financial Aid	5	00:23	0	3.40
North Carolina	2	00:23	0	5.00	Papyrology, palaeography, epigraphy	14	00:36	1	4.92
Library Literature	116	01:20	21	4.65	New York City	1	00:47	0	6.00
Review Material	33	00:32	4	4.94	Classics	24	00:39	0	3.54
Homestead Rebate	1	00:27	1	5.00	Woodcarving	18	00:29	0	2.35
Eyeglasses	7	00:38	1	5.14	Mass Transit	17	00:30	0	3.12
Weddings	53	01:12	1	4.65	Directions	2	00:19	0	3.50
Rescued Beagles	101	00:27	3	5.85	Serials	5	00:56	0	5.20
Poison Ivy	6	02:25	0	4.83	Medieval	6	00:32	0	5.67
Evaluation Criteria	5	00:58	0	4.40	Electronic Resources	1	02:38	0	7.00
Florida	3	00:25	0	5.00	Collection Development	2	00:16	0	5.00
U. of Arizona	4	00:53	0	5.00	Recipe Search	6	05:36	0	4.67
Alexander Library	3	00:15	0	5.67	Evaluation Criteria	3	00:28	1	1.67
Classmate	5	00:16	0	6.20	Book Review Research	37	00:25	1	3.62
Amanda Beasley	5	00:43	0	5.00					
Sailing	9	00:55	0	5.22					
Dog Park	4	00:23	1	4.00					
Radio	3	00:52	0	5.33					
Music	1	00:10	0	4.00					

4 Topic Clustering Techniques

This section contains a description of the various techniques used to cluster the documents viewed by our users, including the rules we used to identify useful documents. This section is followed by a description of the techniques that were used to compare the clusters of documents generated by our users with the clusters generated by the statistical approaches.

We take a language modeling approach to modeling collections of text [5]. Assume we are given a single document as a sample from the urn described in Figure 1. A naïve estimation of the document model would merely count the frequencies of the terms in the document,

$$\hat{P}(w \mid D_i) = \frac{c(w, D_i)}{\sum_{w' \in D_i} c(w', D_i)} \tag{1}$$

where $c(w,D_i)$ represents the number of times word w appears in a particular document, D_i. Unfortunately, if a word does not occur in the document, then its estimated probability will be zero. Since a document is only a sample from this document model, we would like to smooth this estimate with some other model. We accomplish this by interpolating the maximum likelihood document model with a maximum likelihood collection model so that our smoothed document model becomes,

$$P(w \mid D_i) = \lambda \hat{P}(w \mid D_i) + (1 - \lambda)\hat{P}(w \mid C) \tag{2}$$

where *C* is the document collection. In our experiments, was empirically set to 0.90. *Topic* language models can then be constructed by combining the individual document language models. For instance, if we know that a set of documents *T* all discuss the same topic, then we build the topic language models according to,

$$P(w \mid T) = \frac{1}{|T|} \sum_{D_i \in T} P(w \mid D_i) \tag{3}$$

This formalism can be used to build concise summaries of topics by inspecting the language model for each topic. Specifically, we can compute how the topic model, $P(w \mid T)$, differs from the collection model, $\hat{P}(w \mid C)$, by inspecting the *pointwise Kullback-Leibler (KL) divergence* [7,17]. For each word, the pointwise KL divergence is defined as,

$$P(w \mid T) \log \frac{P(w \mid T)}{\hat{P}(w \mid C)} \tag{4}$$

Terms with the highest pointwise KL divergence will be the most discriminating.

These topic models are estimated by using *all* of the evaluated documents and serve not as a method to be evaluated but rather to qualitatively represent the language modeling technique. For example, Table 3 displays the top ten distinguishing words for the topics identified by our users. Notice that it is not necessarily the case that topics with fewer example documents have less meaningful language models. It is more important to have a consistent and precise language. An example of where consistency and precision fail for large topic sizes can be seen in the language model description of User 2's topic "General Interests".

We are interested in automatically recognizing and representing topics in the pages viewed and evaluated by our users. Two clustering methods were implemented for automatically building topic models: k-means clustering and agglomerative clustering.

4.1 K-Means Clustering

In our context, k-means clustering assumes that there are *k* underlying topics responsible for having generated the documents in the training data. The learning process begins by randomly picking *k* documents as cluster representatives or *centroids*. The remaining documents are then assigned to the most similar topic model. For our experiments, similarity is determined by the Kullback-Leibler divergence defined as

Table 3. Language model (LM) descriptions of user topics

User 1		User 2	
Topic	LM Description	Topic	LM Description
Theology	Biblical, bible, Israel, theology	General Interest	Weather, jesus, movie, film, time, home
Perennials	Geranium, plant, garden, big	College Financial Aid	Loan, pay, hesc, forbearance, borrow
North Carolina	Weather, forecast, low, high	Papyrology, palaeography, epigraphy	Citation, library, abstract, full, article
Library Literature	Library, information, service	New York City	Square, greenmarket, union, park
Review Material	Title, library, book, footage	Classics	Classic, rate, site, resource, ancient
Homestead Rebate	Taxation, treasury, rebate, state	Woodcarving	Nantucket, art, carve, stbart, gallery
Eyeglasses	Store, lenscrafter, offer, rate	Mass Transit	Transit, rail, corridor, transportation
Weddings	Wedding, bride, indiebride, club	Directions	Switchboard, starbuck, map, search
Rescued Beagles	Pet, petfinder, beagle, dog, org	Serials	Edit, reprint, und, von, teil, die, der
Poison Ivy	Hive, webmd, cause, post	Medieval	Der, kehr, papsturkunden, paul, fridolin
Evaluation Criteria	Evaluate, internet, site, web	Electronic Resources	Rom, text, edition, database, English
Florida	Mapquest, map, flight	Collection Development	Record, franco, view, gesta, dei, nogent
U. of Arizona	Semester, session, class, summer	Recipe Search	Recipe, chicken, epicurious, cook, sauce
Alexander Library	Rutgers, library, alex, summer	Evaluation Criteria	Hon, honcode, medical, health
Classmate	Yahoo, map, locate, glen, address	Book Review Research	Book, review, booklist, june
Amanda Beasley	Elliot, ilisha, nerve, feature		
Sailing	Race, Bermuda, Newport		
Dog Park	Maplewood, construct		
Radio	Wfuv, wnyc, folk, stream, city		
Music	Garbage, 22garbage, band, google		

$$KL(D_i \,\|\, T) = \sum_{w \in V} P(w \mid D_i) \log \frac{P(w \mid D_i)}{P(w \mid T)} \qquad (5)$$

where V is the vocabulary. After all documents have been assigned, topic models are then re-estimated using the new document topic sets. This assignment and estimation process continues until topic models converge. For each user, k was set to the known number of topics.

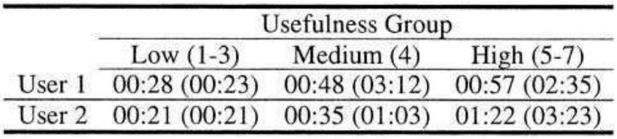
Selection of Seed Documents. We assume that in real interaction there will be documents already associated with particular topics. Therefore, we next consider how to feed examples to this algorithm. Our approach is to assign these examples to the initial centroids and fix these assignments throughout the execution of the learning. Therefore, the example documents will always be a component of the topic model.

The method of selection for these seed documents consisted of identifying documents receiving the highest usefulness rating (7) and the highest confidence rating (7) by our users for each topic. In cases where more than one document met the selection criteria, an attempt was made to select documents that were viewed on different days. In cases where only a single day was represented, the first two documents meeting the criteria were selected. If there were no documents for a specific topic class with a usefulness rating of 7, then documents that received a 6, the second highest usefulness rating, were selected. In most cases the confidence scores were always high, so it was possible to select documents which had high confidence scores associated with them.

Display Time and Retention. In addition to providing seeds for the topic models, we also considered weighting documents depending on their import. In particular, we were interested in using the display time and retention behaviors of our users as implicit evidence of usefulness. The goal in doing this was to unobtrusively identify documents whose weight could be increased during the automatic clustering process.

For many topics, there were only a few documents viewed. Because of this, we used a measure of display time based on the overall display times and usefulness ratings for each user, rather than those display times observed for individual topics. We grouped the points of our 7-point usefulness scale into 3 classes: Low (1-3), Medium (4) and High (5-7). We then computed the mean display time for each of these groups for each user and used the mean display time for the high usefulness group as a method for identifying useful documents. Thus, if a user displayed a document for longer than this mean display time, then the weight of this document was increased during clustering. The means for each usefulness group are displayed in Table 4. Our use of retention was a little more straightforward. If a retention behavior occurred at a document (i.e. the user printed, saved, emailed or bookmarked the document), then we used this to increase the weight associated with the document during clustering.

Table 4. Mean and standard deviation display times according to usefulness



	Usefulness Group		
	Low (1-3)	Medium (4)	High (5-7)
User 1	00:28 (00:23)	00:48 (03:12)	00:57 (02:35)
User 2	00:21 (00:21)	00:35 (01:03)	01:22 (03:23)

4.2 Agglomerative Clustering

One drawback to the k-means approach is the requirement that we know the number of topics, *k*. As an alternative, we also evaluated two agglomerative clustering techniques. Agglomerative clustering techniques build topic representations bottom up. The algorithm begins with each document in its own cluster and then successively merges clusters according to similarity. The method always merges the two closest clusters. It is the interpretation of *closest* which differentiates our two agglomerative techniques. In both cases, clustering terminates when the similarity between the closest clusters is below a certain threshold.

Single-link Clustering. One possible interpretation of inter-cluster distance considers the shortest distance between all inter-cluster document pairs (i.e. a document belongs to the same topic as its most similar neighbor). For this algorithm to be consistent, we use the J-divergence, a symmetric version of the KL-divergence measure,

$$J(D_i \parallel D_j) = KL(D_i \parallel D_j) + KL(D_j \parallel D_i) \tag{6}$$

It is important to notice that the single-link technique provides no explicit representation of a topic model. Because of this, a method for seeding the algorithm with topic examples is not obvious and was not used for this technique.

Average-link Clustering. Although single link clustering performs well in traditional topic tasks, it has a tendency to create topic models covering a variety of sub-topics; this is a product of a document only needing a single highly similar match to be included in the cluster. Instead, we may want to assign a document to the cluster to which it has the highest *average* similarity. In this case, the similarity between two clusters is calculated by averaging the similarity between all pairs of documents between two clusters.

5 Evaluation Techniques

We used as ground truth the clusters that resulted from our users' classification of the documents that they viewed into self-identified topics. The automatically-generated clusters were evaluated by measuring the accuracy of predicted links between documents. That is, two documents in the same cluster are said to have a link between them. If there are N documents for a particular user, then there are $O(N^2)$ possible links between all pairs of documents. Let this total set be L. Let the set of true links defined by the manual clustering of the documents be defined by $L' \subseteq L$. Let the set of links predicted by the system be define by $L_s \subseteq L$. We evaluate the performance of our systems using two measures of accuracy. First, we measure the *total accuracy* of prediction,

$$\frac{|L' \cap L_s|}{|L'|} + \frac{|\overline{L'} \cap \overline{L_s}|}{|\overline{L'}|} \tag{7}$$

This evaluates the system prediction of link presence and absence in a set of documents. Our second measure focuses on the accuracy of predicting true links. Specifically, we use the equation,

$$\frac{|L' \cap L_s|}{|L'|} \tag{8}$$

which will provide a means for disambiguating the degree to which good *total accuracy* relies upon keeping unrelated documents in separate clusters. We will refer to this as *link recall.*

6 Results

Eight variants of the k-means clustering technique were used which incorporated different degrees of evidence and re-weighting. The agglomerative techniques were run without any evidence or re-weighting. Thresholds for the clustering were empirically set. Table 5 presents the total link accuracy and link recall for each subject, for each technique. The results presented in Table 5 indicate that seeding clusters provides valuable information for the k-means techniques. In all cases, seeded clusters out-perform the unseeded counterparts. However, other results for the k-means techniques are less conclusive. For example, the effect of re-weighting schemes such as display time on performance is mixed. We speculate that a more sophisticated incorporation in the k-means model might provide better results. Surprisingly, the knowledge-poor agglomerative techniques performed as well or better in three out of the four trials.

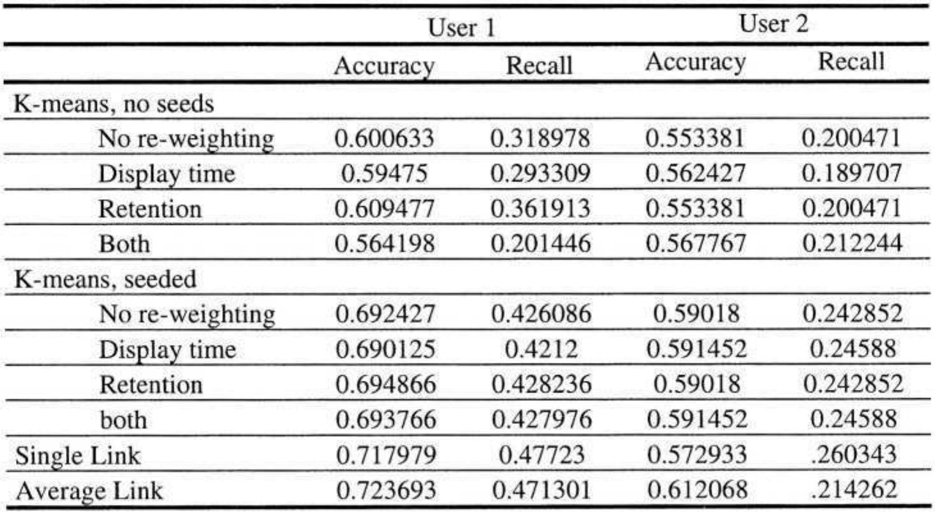
While these results provide gross estimates of system performance, we would like to measure the actual number of true links retrieved. Table 5 displays the accuracy of predicting topical links between documents (link recall). Again, seeding and re-weighting improve performance. Further, the agglomerative techniques perform better than k-means for User 1. The agglomerative results for User 2 are less conclusive perhaps as a result of the small collection size.

We speculate that the poor performance of the k-means experiments for User 1 is the result of fixing *k.* If the language of the documents does not follow a topical pattern, then restricting the potential cluster assignments will result in conflating distantly related documents. The agglomerative techniques are quite content leaving those outliers as singleton clusters, effectively remaining agnostic about topic assignment. This is confirmed by the large number of singleton clusters in the agglomerative techniques. These statistics are shown in Table 6.

In order to further test this hypothesis, additional k-means experiments were performed with alternate values for *k.* No seeded experiments were performed because there would be fewer seeds than clusters. After sweeping a range of *k* from 21 to 50, a value of 30, in general, improved the unseeded performance the most compared to the original experiments. The results for these experiments are shown in Table 7. Note that only the un-weighted and display time-weighted techniques actually improved with an increase in *k.* In fact, the performance of methods incorporating retention is, in general, worse when we increase the number of clusters.

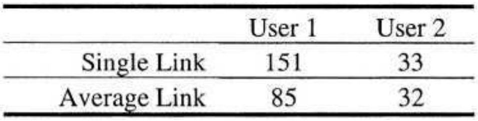
These results indicate that the agglomerative techniques are more successful for User 1 not because of their superior representation but rather because they take fewer risks in deciding that two documents are on the same topic. The k-means algorithms, on the other hand, are potentially forced to take these risks. The benefit is that if the language encodes the topics, accuracy of known links is better than the agglomerative techniques.

Table 5. Total link accuracy and link recall



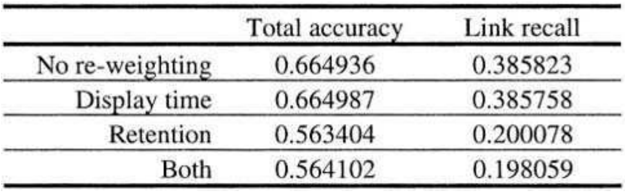
	User 1		User 2	
	Accuracy	Recall	Accuracy	Recall
K-means, no seeds				
No re-weighting	0.600633	0.318978	0.553381	0.200471
Display time	0.59475	0.293309	0.562427	0.189707
Retention	0.609477	0.361913	0.553381	0.200471
Both	0.564198	0.201446	0.567767	0.212244
K-means, seeded				
No re-weighting	0.692427	0.426086	0.59018	0.242852
Display time	0.690125	0.4212	0.591452	0.24588
Retention	0.694866	0.428236	0.59018	0.242852
both	0.693766	0.427976	0.591452	0.24588
Single Link	0.717979	0.47723	0.572933	.260343
Average Link	0.723693	0.471301	0.612068	.214262

Table 6. Number of clusters generated by agglomerative techniques



	User 1	User 2
Single Link	151	33
Average Link	85	32

Table 7. 30-means experiments for User 1, no seeds



	Total accuracy	Link recall
No re-weighting	0.664936	0.385823
Display time	0.664987	0.385758
Retention	0.563404	0.200078
Both	0.564102	0.198059

7 Discussion

For both users, the average-link technique provided the most accurate performance. The k-means technique without seeds performed the worst, even when display time and retention were considered. In all cases, seeded clusters out-performed the unseeded counterparts and models of User 1's topics were more accurate than User 2's. Including behavioral evidence for re-weighting documents resulted in little, if any, improvement. When considering link recall, the agglomerative techniques outperformed the k-mean techniques and the seeded clusters still outperformed the unseeded.

The use of display time and retention had little effect on clustering with or without seed documents. It may be the case that our measure of identifying useful documents based on display times was not the most effective. Previous work [11] has found that while documents that are rated more highly usually have higher mean display times, they also have higher variance, which might make it difficult for measures of mean display times to perform very well. An examination of Table 4 demonstrates that this may be the case for this data as well. Thus, our measure of usefulness based on display time may not have been selective enough. Moreover, User 1 rated a large portion of the documents that she viewed as 5 or 6, which are points included in the high usefulness group. While our use of mean display time is an improvement over previous work since its computation is based on the behavior of each individual user as opposed to a group of users, it still may not have been sensitive enough.

Retention was found to increase performance slightly for User 1 when no seeds were used (Table 5), perhaps indicating its potential as a technique for identifying documents that could be used as seeds or for re-weighting. User 1 exhibited more retention behaviors (33 documents) than User 2 (7 documents), which may explain why clustering for User 1 benefited from the inclusion of retention for re-weighting.

Set size and quality may have also affected our results. In terms of total number of documents evaluated, User 1 viewed and evaluated considerably more than User 2, while only identifying 5 more topics. It should be noted again that our users did not evaluate all of the documents that they viewed during the 5-week period. Instead, we screened documents using a classification scheme to eliminate email pages, advertisements, discussion groups and search pages. Thus, we believe that the quality of the documents that were evaluated and used in the clustering, were better than if we had used all displayed documents. Although we cannot be certain without conducting the analysis, clustering only the set of documents that users evaluated most likely resulted in more accurate topic models than clustering all displayed documents.

The number of documents users associated with each topic varied considerably. For some topics, 50 or more documents were associated with the topic, but it was more often the case that a large number of topics had 5 or fewer documents associated with them. Given that we used 2 seed documents per topic, it is unsurprising that the k-means with seeds out-performed no seeds for many topics.

8 Conclusions

Overall, the techniques we used for topic model construction performed poorly when evaluated according to the user-defined topic classes. It is unclear if automatic clustering techniques can be as sensitive as users when creating and assigning documents to topic clusters. However, we feel that more attempts at user-centered approaches to the evaluation of topic models are necessary and that the clusters created by users can provide an evaluation metric of the highest standard. Moreover, great care was invested in developing the methodology used in our monitoring study, and we believe that this methodology can act as a valuable model for others interested in exploring user-centered approaches to evaluating automatically-generated topic models.

A combination of the quality of the sets and the seeds seems to play a significant role during clustering. The use of seeds improved performance for both users and suggests that the identification of quality seeds may be necessary for accurate topic modeling. Additionally, the quality of the documents that were evaluated and used in the clustering were better than if we had used all displayed documents. Certainly the definition of a “quality” document is rather nebulous and more work needs to be done understanding and identifying the attributes of quality documents. If we are to create a system that makes use of a user’s web browsing history, then the system needs to know when it should consider a document for inclusion in topic clustering. Whether relevant or not relevant, not all documents are equally useful in constructing topic models. A document that only contains a search box is not as useful as one which contains the text of a conference paper. This also applies to using behavior as implicit feedback: observing a high display time at a document containing a search box and little text most likely indicates something different than observing a high display time at document containing a conference paper. Clearly, the system needs some assistance in identifying candidate documents for inclusion in topic modeling and as sources of implicit feedback. We are currently working to develop our web page classification for use in future experiments and hope that this will elucidate some aspects of “quality”documents and their impact on modeling.

We have just finished a second naturalistic study of the sort described in this paper with seven new users, which lasted 3.5 months. We plan to conduct an analysis and evaluation similar to the one described in this paper and adjust our display time measure, as well as investigate the usefulness of the task groupings and additional behavioral data in the construction of topic models. Ultimately, we would like to use these models to provide personalized information retrieval to individuals.

Acknowledgments. This work was partially supported by the CIIR, NSF Grant #IIS-9907018, and NSF Grant #99-11942. Opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor. We would like to thank Doug Riecken and IBM for the loan of the laptops and Haizheng Zhang for the proxy logger.

References

1. Allan, J., (Ed.) Topic Detection and Tracking, Event-based Information Organization, Kluwer Academic Publishers, 2002.
2. Allan, J., Lavrenko, V., & Swan, R. Explorations within topic tracking and detection. In Topic Detection and Tracking, Event-based Information Organization (James Allan, Ed.), Kluwer Academic Publishers, 197-224, 2002.
3. Bhatia, S. K. & Deogun, J. S. Cluster Characterization in Information Retrieval. Proceedings of SIGIR ’93, 721-728, 1993.
4. Claypool, M., Le, P., Waseda, M., & Brown, D. Implicit interest indicators. Proceedings of Intelligent User Interfaces (IUI ’02), 2001.
5. Croft, W.B. & Lafferty, J., (Eds.) Language Modeling for Information Retrieval, Kluwer Academic Publishers, 2003.
6. Deogun, J.S. & Raghavan, V.V. User-oriented Document Clustering: A framework for learning in information retrieval. Proceedings of the ACM SIGIR ’86, 157-163, 1986.

7. Diaz, F. & Allan, J., Browsing-based User Language Models for Information Retrieval, CIIR Technical Report IR-279, 2003.
8. Gordon, M. User-based document clustering by redescribing subject descriptions with a genetic algorithm. JASIST, 42, 311-322, 1991.
9. Heer, J., & Chi, E. H. Separating the swarm: Categorization methods for user sessions on the web. Proceedings of CHI '02,243-250, 2002.
10. Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L.R. & Riedl, J. GroupLens: Applying collaborative filtering to Usenet news. Communications of the ACM, 40(7), 77-87,1997.
11. Kelly, D. & Belkin, N. J. Reading time, scrolling and interaction: Exploring implicit sources of user preferences for relevance feedback during interactive information retrieval. Proceedings SIGIR '01, 408-409, 2001.
12. Morita, M., & Shinoda, Y. Information filtering based on user behavior analysis and best match text retrieval. Proceedings of SIGIR '94, 272-281, 1994.
13. Oard, D. W., & Kim, J. Modeling information content using observable behavior. Proceedings of ASIST '01, 38-45, 2001.
14. Pazzani, M. & Billsus, D. Learning and revisiting user profiles: The identification of interesting web sites. Machine Learning 27, 313-331, 1997.
15. Ruthven, I., Lalmas, M., van Rijsbergen, K. Incorporating user search behavior into relevance feedback. JASIST, 54, 529-549, 2003.
16. Seo, Y. W., & Zhang, B. T. Learning user's preferences by analyzing wed-browsing behaviors. Proceedings of Autonomous Agents, 381-387, 2000.
17. Tomokiyo, T. & Hurst, M. A language model approach to keyphrase extraction. Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition & Treatment, 33-40, 2003.
18. Zhai, C., Cohen, W.W., & Lafferty, J. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. Proceedings of SIGIR '03, 10-17, 2003.

A Study of User Interaction with a Concept-Based Interactive Query Expansion Support Tool

Hideo Joho, Mark Sanderson, and Micheline Beaulieu

Department of Information Studies, University of Sheffield,
Regent Court, 211 Portobello Street, Sheffield, S1 4DP.
{H.Joho,M.Sanderson,M.Beaulieu}@sheffield.ac.uk

Abstract. A medium-scale user study was carried out to investigate the usability of a concept-based query expansion support tool. The tool was fully integrated into the interface of an IR system, and designed to support the user by offering automatically generated concept hierarchies. Two types of hierarchies were compared with a baseline. Several observations were made as a result of the study: 1) the hierarchy is often accessed after an examination of the first page of search results; 2) accessing the hierarchies reduces the number of iterations and paging actions; 3) accessing the hierarchies increases the chance of finding relevant items more accurately than the baseline; 4) the hierarchical structure helps the users to handle a large number of concepts; and finally, 5) subjects were not aware of the difference between two types of hierarchies.

1 Introduction

In interactive query expansion (IQE), users often find it difficult to select expansion terms from a suggested list [1,2]. Possible reasons for this is that the statistical weighting tends to generate low frequency, specific, or unfamiliar terms, and the list does not provide the context for the suggested terms. However, our previous study and others suggest that the hierarchical organisation of candidate expansion terms can offer better both context and greater efficiency in the query expansion process [3,4]. This paper presents a user study of a concept-based approach to IQE.

CiQuest (Concept-based Interactive QUery Expansion Support Tool) is a support system for interactive searches. It provides an overview of a set of retrieved documents which allows the user to focus on a particular subset of the search results. It also provides a set of candidate terms that can be used to replace or expand a user's initial query. The CiQuest system is designed to achieve these two facilities through concept hierarchies. A concept hierarchy is *dynamically* generated from a set of retrieved documents and visualised by cascading menus. More general terms are placed at a higher level followed by related but more specific terms at a lower level.

Our overall research aim is to study the use of a concept-based system to support information retrieval. The specific objectives are to:

- evaluate the retrieval effectiveness of document derived concept structures for selecting relevant documents in a retrieved document set;
- evaluate the retrieval effectiveness of incorporating concept structures to assist users in selecting candidate terms for interactive query expansion; and

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 42–56, 2004.

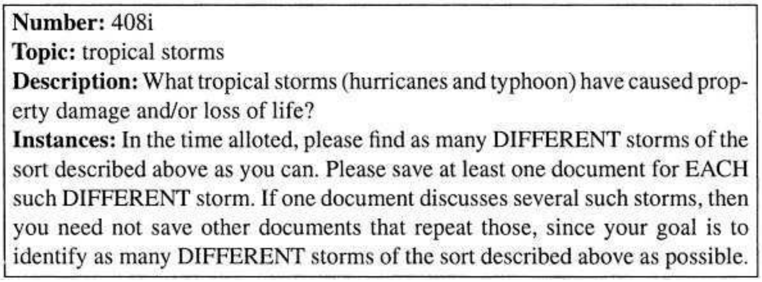
- assess how searchers make use of concept structures to bridge the gap between the query space and the document space in interactive searching.

The next section will discuss our experimental methodology including the details of our system and experimental design. The results and analysis of our experiments will then be presented. The paper concludes with an overall discussion of our findings and future work.

2 Experimental Design

The Interactive Track of TREC (Text REtrieval Conference[1]) has been developing a test collection for research into interactive information retrieval. We used the test collection from TREC-8 Interactive Track [5] as well as the Ad-hoc task as the basis of our experiments. It consists of six topics, relevance information, and a collection of 210,158 articles (564MB of texts) from the Financial Times 1991-1994. Each topic contains a title, description, and definition of instances as shown in Fig. 1.

The task defined by the TREC-8 Interactive Track is referred to as an *instance finding task*. In this task the subjects are asked to find as many different instances or answers to the query as possible, as opposed to finding as many relevant documents as possible as in the Ad-hoc task. For example, Topic 408i is designed to find the instances of the tropical storms that have caused property damage or loss of life. The subjects are also asked to save at least one document for each of the different aspects or answers of the topic.



Number: 408i
Topic: tropical storms
Description: What tropical storms (hurricanes and typhoon) have caused property damage and/or loss of life?
Instances: In the time alloted, please find as many DIFFERENT storms of the sort described above as you can. Please save at least one document for EACH such DIFFERENT storm. If one document discusses several such storms, then you need not save other documents that repeat those, since your goal is to identify as many DIFFERENT storms of the sort described above as possible.

Fig. 1. TREC-8 Interactive Track Sample Topic (408i)

2.1 Participants

Twelve participants were recruited from Department of Information Studies and Computer Science, and included two females and ten males who were either research students or research assistants. Their educational qualification included one with a PhD, eight with a Master, and three with a Bachelor. Of the twelve, two had participated a TREC experiment before but neither had experience of seeing the topics and tasks used in our experiment.

[1] http://trec.nist.gov/

2.2 System and Interface Development

The CiQuest system is a tool designed to support information access through two basic functionalities: multi-document summarisation and interactive query expansion. Words and noun phrases (i.e. *concepts*) are extracted from the retrieved documents and used to form a hierarchical structure which, as a whole, can be seen as a summary of search results. Individual concepts that are organised in a *general to specific* manner and can also be seen as candidate terms to expand or reformulate initial queries.

The core technology of the system is to determine the semantic specificity of concepts with little human involvement or knowledge resources. Our overall aim is to find a pair of related concepts and determine which is more general (or specific). A hierarchy is, thus, formed as a result of the cumulation of such a process. For our experiment we have implemented two different approaches for the generation of the hierarchies.

Generating hierarchies. The first approach is based on the statistical analysis of document frequency and co-occurrence information between concepts, and called the subsumption approach which was originally developed by Sanderson and Croft [6]. In this approach, concept C_i is said to subsume concept C_j when a set of documents in which C_j occurs is a subset of the documents in which C_i occurs, or more specifically, when the following two conditions are held: $P(C_j \mid C_i) \geq 0.8$[2] and $P(C_i \mid C_j) < 1$.

The assumption is that C_i is likely to be more general than C_j because, first, the former appears more frequently than the latter, and second, the former subsumes a large part of C_j's document set. Also they are likely to be related since they co-occur frequently within documents. A similar assumption has been made by other researchers (e.g. [7,8]. A sample hierarchy using this approach can be found in Fig. 2.

The second approach is called the trigger phrase approach, and is based on the lexical and syntactic analysis of noun phrases which have been found to be useful for query expansion [9]. A trigger phrase is a phrase that matches a fragment of text that contains a parent-child description. Words and phrases found in the description are used to formulate the hierarchy. Our trigger phrases are based on Hearst [10] who originally used them to find additional lexical relations in WordNet [11]. Examples of the phrase patterns are:

- SUCH AS: ... international organisations **such as** WHO, NATO, and ...
- AND OTHER: ... WHO, NATO, **and other** international organisations are ...
- INCLUDING: ... international organisations, **including** WHO, NATO, and ...

In the above example, when one of the patterns is matched, the concept *international organisations* is set as a superordinate of *WHO* and *NATO* in the above example. Furthermore, the head noun of phrases is identified and set as a superordinate of the phrases (similar to [12]). For example, *organisations* (head noun) is set as a superordinate of *international organisations*. This head noun extraction also helps the hierarchy to include more phrases that contains the same head noun. In other words, this approach attempts to generate a hierarchy of noun phrases using the lexical evidence and the head nouns. A sample hierarchy using this approach can be found in Fig. 3.

[2] This value was set by them empirically.

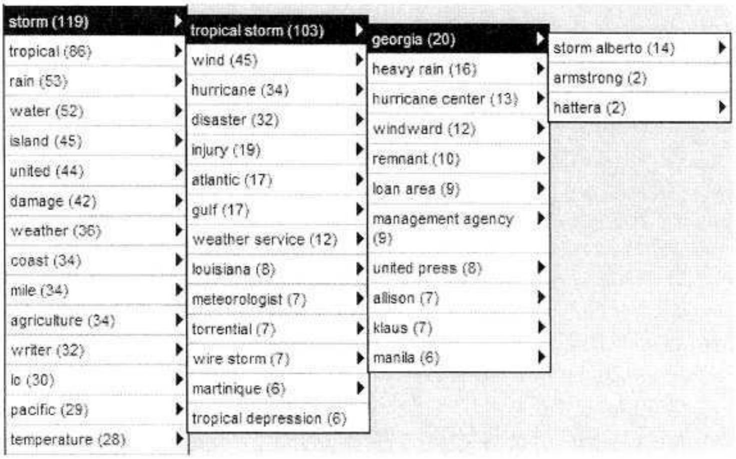


Fig. 2. Sample hierarchy generated by the subsumption approach with the top 200 documents retrieved in response to the query *tropical storm.* The number next to the term indicates the frequency of occurrence. You can see the phrase "tropical storm" is subsumed by the term "storm". Also several instances of storms or hurricanes such as *george, allison,* or *klaus* are successfully organised under "tropical storm".

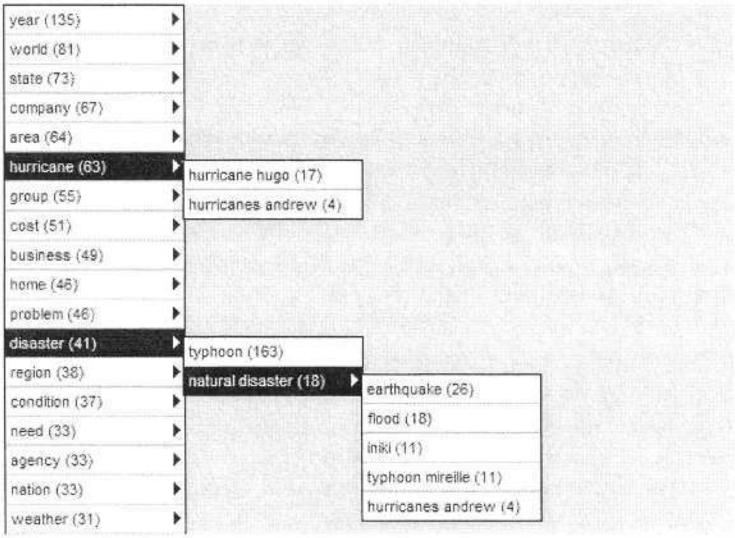
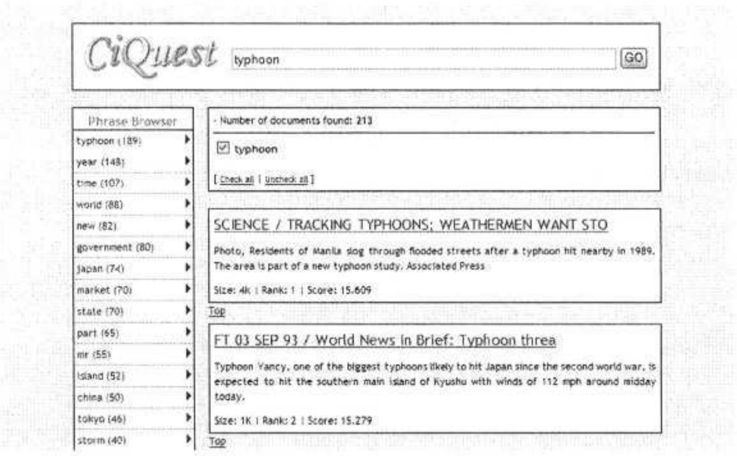


Fig. 3. Sample hierarchy generated by the trigger phrase approach with the top 200 documents retrieved in response to the query *typhoon hurricane.* Noun phrases such as *hurricane hugo* and *hurricane andrew* can be found under the head noun "hurricane" at the top level of the hierarchy. Also you can find the terms such as *earthquake, flood,* and phrases including *typhoon* or *hurricane* organised as an instance of "natural disaster".

CiQuest system in use. Once a hierarchical structure of related concepts is generated, the system visualises it using cascading menus. The top level of hierarchies are shown in the left side of the main result page (See Fig 4). Our principle regarding the integration of the hierarchy into an IR system's interface is to provide the functionality without disturbing the default search process. The default search process is to submit a query, look through the hitlist, and open a page to access the fulltext.



· Number of documents found: 213

☑ typhoon

SCIENCE / TRACKING TYPHOONS; WEATHERMEN WANT STO

Photo, Residents of Manila slog through flooded streets after a typhoon hit nearby in 1989. The area is part of a new typhoon study. Associated Press

Size: 4k | Rank: 1 | Score: 15.609

FT 03 SEP 93 / World News in Brief: Typhoon threa

Typhoon Yancy, one of the biggest typhoons likely to hit Japan since the second world war, is expected to hit the southern main island of Kyushu with winds of 112 mph around midday today.

Size: 1K | Rank: 2 | Score: 15.279

Fig. 4. CiQuest system: Top level of menu is shown along with the search result

Backend IR system: CiQuest system in the current paper was integrated into the Okapi system [13]. The best passage identified by the weighting scheme was displayed in every record of search results.

Browsing the hierarchy: When a mouse pointer is *placed* on a concept in the menu, a list of its subordinate concepts is displayed. The presence of subordinates is indicated by a small triangle arrow at the right-side of each entry.

Focusing on a subset: When a concept in the menu is *clicked,* a set of documents in which the concept occurs within the retrieved documents is shown in the same format as in the initial results. This subset of documents is also ordered by the ranking of the initial results. In this *focusing mode,* a pointer link is displayed at the bottom of the page to allow the user to go back to the initial results.

Refreshing the hierarchy: When another query is submitted, the hierarchy is automatically refreshed based on a set of documents retrieved in response to the new query.

2.3 Experimental Procedures

Experiments were based on the CiQuest system, but three different versions were devised for the test. The first was a baseline system which offered no support function. The second and third versions each incorporated the subsumption and the trigger phrase approaches respectively. Although the underlying functionality was different, subjects were not made aware of this as they searched through a common web-based interface.

Each test subject undertook searches on three TREC-8 topics, one to test each version of the system. The allocation of topics and test system was done randomly so that each topic was, thus, searched by six subjects. Participants were briefed on two tasks: the first was the *instance finding* tasks as described above. The second task, *query optimising,* required searchers to generate a so-called optimal or best query based on their search experience of the topic. The optimising task made it possible to compare the effectiveness of the optimal query with that of the initial query based on precision and recall for document relevance as used for the TREC Ad-hoc task, as opposed to the instance relevance used in the Interactive task.

The first exeriment, therefore, is the true interactive searching task, and the second experiment is a black-box input/output approach which does not take account of user interaction.

After the demonstration of the system, subjects were given several minutes to use the system with a sample topic. The subjects were then given 10 minutes for the instance finding task, but were allowed to take as long as they wish for the query optimising task. However, they tended to complete the task within a couple of minutes. Subjects also completed questionnaires at the beginning of the test session, after each search, and on completing the whole experiment. The questionnaires were based on the instruments developed for the TREC Interactive Track. The procedure took 60 to 90 minutes in total for each subject.

3 Results and Analysis

The results and analysis of our experiments using the precision/recall measures[3], log analysis, questionnaire, and manual observation are as follows. Three groups of the system settings as described above will be referred to as the *Baseline, Subsump* menu, and *Trigger* menu in this section.

3.1 Instance Finding Task

Instance recall and precision. The instance recall is calculated based on the number of instances correctly identified by the subjects divided by the total number of instances identified by the NIST assessors (called official instances). The instance precision is calculated based on the number of correctly identified instances divided by the total number of instances identified by the subjects.

Table 1 shows the instance recall and precision of the three groups. Each topic was used by two subjects in all groups. Although the difference among the groups are generally small, the result shows that the Baseline's recall is higher than the menu groups while the Subsump achieved the highest precision among them.

As for the higher recall with the Baseline, two reasons can be possible. One is that the Okapi back-end IR system performed well [14], thus, the subjects could find relevant

[3] Overall, it was rare to find the statistical significance using t-test due to the sample size, but it is indicated by a star (*) where applicable.

Table 1. Instance recall and precision

		Baseline		Subsump		Trigger	
	Official	Instance	Instance	Instance	Instance	Instance	Instance
Topic ID	instances	recall	precision	recall	precision	recall	precision
408i	24	0.313	0.834	0.250	0.659	0.084	0.667
414i	12	0.375	0.729	0.292	0.875	0.459	0.745
428i	26	0.423	0.816	0.289	0.917	0.231	0.709
431i	40	0.138	0.625	0.113	0.625	0.175	0.399
438i	56	0.215	0.690	0.188	0.857	0.161	0.988
446i	16	0.188	0.715	0.282	0.700	0.313	0.410
Average		0.275	0.735	0.235	0.772	0.237	0.653

instances without support. Another is that the Baseline group could spent more time to examine a greater number of documents while the menu groups were spending the time browsing the hierarchies. However, the higher precision with the Subsump suggests that the accuracy of identifying relevant instances can be improved by the hierarchies.

Document access rate. The subjects were asked to save a document in which they found one or more instances. Table 2 shows the number of documents in which subjects selected and viewed the full-text (called seen documents) and documents that were actually saved as relevant.

Table 2. Document access rate (%)

	Base			Subsump			Trigger		
Topic ID	Seen doc	Saved doc	Rate	Seen doc	Saved doc	Rate	Seen doc	Saved doc	Rate
408i	18.5	9.0	52.58	16.5	8.0	47.98	10.5	2.5	26.44
414i	11.0	4.5	40.00	6.0	2.5	41.43	10.5	3.5	36.12
428i	14.0	11.0	80.75	13.5	9.0	66.49	9.0	7.0	77.78
431i	17.5	8.5	49.02	11.5	7.5	67.50	8.0	4.5	73.08
438i	23.5	17.0	72.64	12.0	11.0	92.86	11.0	9.0	83.04
446i	13.5	5.0	38.93	12.5	6.0	47.73	12.0	10.5	87.77
Average	16.3	9.17	55.65	12.0	7.3	60.66	10.2	6.2	64.04

As can be seen, the subjects viewed more documents in the Baseline than the Subsump or Trigger but saved less frequently. With the menus the seen documents were more often saved. Here, with the previous table's result, we can see a trend of improving the accuracy of identifying relevant documents and instances when the hierarchies were used.

Interaction, paging, and access to the menu. Table 3 shows the data about the iteration of searches, paging, and access to the menus, which provides additional insight of user behaviour in the instance finding task. An iteration is defined as a new query or refomulated query in the course of session. A paging is defined as moving one result

Table 3. Iteration, paging, and access to the menu

	Base		Subsump				Trigger			
Topic ID	Iter	Paging	Iter	Paging	Menu	Saved	Iter	Paging	Menu	Saved
408i	6.5	7.0	3.5	2.0	8.0	2.5	8.5	0.0	3.5	0.0
414i	5.5	5.5	2.5	2.0	4.5	0.5	3.0	1.5	4.0	1.0
428i	3.0	5.0	2.0	1.5	4.5	2.5	1.5	2.0	1.5	0.5
431i	4.5	2.0	4.0	1.5	0.5	0.0	4.0	2.5	3.0	1.0
438i	4.5	2.0	4.0	1.5	0.5	0.0	4.0	2.5	3.0	1.0
446i	3.0	4.5	3.0	1.5	6.0	0.5	3.5	4.0	1.0	0.0
Average	4.50	4.33	3.17	1.67	4.00	1.00	4.08	2.08	2.67	0.58

page to another. A menu access is defined as clicking on a concept term to display the set of linked documents.

First, the number of iterations shows that the subjects submitted fewer queries with the menu groups than the Baseline. Also, the frequency of going to the next page in the Baseline is higher than the menu groups. Both, along with Menu access information, indicate that the menus were used to focus on a subset of documents as opposed to submitting a new query or going to the next pages. Saved access is the number of accesses to the menus which lead to save any documents (i.e. find an instance). In this regard, it appears that the Subsump performed marginally better than the Trigger menu.

Summary. Overall, the results from the instance finding task suggests that the menus can be useful to accurately identity relevant information from search results, and reduce the number of iterations and paging actions (i.e. takes less effort).

3.2 Query Optimising Task

The query optimising task was evaluated using the relevance judgements of the TREC-8 Ad-hoc task. The purpose of this task was to compare the effectiveness of the optimal query with that of the initial query based on precision and recall for the full retrieved document sets, as opposed to the documents viewed and judged by the subjects.

Overall. Table 4 shows the retrieval effectiveness of initial queries, which are the first query submitted by the subjects, and optimised queries, which the subject generated after searching each topic. This result confirmed that the subjects could improve their initial queries after 10 minutes of search experience.

Out of 36 sessions, 32 initial queries were modified and four were unchanged. Out of 32 changed queries, 20 had an increase of terms, 6 had a decrease, and 6 had no difference in number. The number of increased terms varies between one and three with the average of 1.45 terms. Although the overall changes against the initial queries were small, As can be seen in Table 4, these small changes contributed to the retrieval of a significantly larger number of relevant documents.

Table 4. Overall performance of query optimisation

	Initial	Optimised	Diff.(%)
No. of session	36	36	
No . of Retrieved Rel docs	2512	3050	21.42*
Precision			
At 1 docs	0.5278	0.6111	15.80
At 5 docs	0.5333	0.5611	5.20
At 10 docs	0.4472	0.5056	13.00
At 20 docs	0.4069	0.4569	12.30
At 30 docs	0.362	0.3981	10.00
Avg. Prec	0.2029	0.2348	15.72

* indicates statistical significance at $p < 0.05$

Table 5. Query optimisation across the systems

	Baseline			Subsump			Trigger		
	Initial	Opt.	Diff.(%)	Initial	Opt.	Diff.(%)	Initial	Opt.	Diff.(%)
No. of session	12	12		12	12		12	12	
Retrieved Rel	781	979	25.35	923	1033	11.92	808	1038	28.47
Precision									
At 1 docs	0.500	0.583	16.70	0.583	0.667	14.30	0.500	0.583	16.70
At 5 docs	0.517	0.550	6.50	0.517	0.533	3.20	0.567	0.600	5.90
At 10 docs	0.450	0.450	0.00	0.425	0.492	15.70	0.467	0.575	23.20
At 20 docs	0.396	0.425	7.40	0.400	0.454	13.50	0.425	0.492	15.70
At 30 docs	0.347	0.361	4.00	0.381	0.408	7.30	0.358	0.425	18.60
Avg. Prec.	0.195	0.224	14.97	0.208	0.228	10.01	0.206	0.252	22.17

Across the system setting. Table 5 shows the comparison of initial and optimised queries over the three system settings. As expected the performance of initial queries were found to be similar across the systems and they were lower than the optimised queries. However, based on the previous task, we did not expect the Trigger menue session to outperform others. From the average precision we can see the Trigger menu contributed most in generating a better query, followed by the Baseline, and Subsump.

Across the topics. Table 6 shows the retrieval effectiveness of both types of queries over six topics used in our experiment. Overall, the optimised queries outperformed the initial ones in all topics with the exception of Topic 408i.

An interesting point is that the improvement achieved by the optimised queries seems to be reasonably consistent across topics which had varied performances of the initial queries (e.g. from 0.1070 to 0.3383 in Average Precision). Although more data would be required to draw any conclusive comments, it seems that the optimised queries could improve the retrieval effectiveness regardless of the performance of initial results.

Summary. The results from the query optimising task shows that the learning curve for optimising their initial queries are similar among the three groups. However it appears that the Trigger group performed marginally better than the other two groups. The

Table 6. Query optimisation across the topic

Topic	408i			414i			428i		
	Initial	Opt.	Diff (%)	Initial	Opt.	Diff (%)	Initial	Opt.	Diff (%)
No. of session	6	6		6	6		6	6	
Retrieved Rel	379	320	-15.57	212	188	-11.32	525	614	16.95
Precision									
At 5 docs	0.333	0.167	-50.00	0.500	0.567	13.30	0.633	0.700	10.50
At 10 docs	0.250	0.167	-33.30	0.367	0.500	36.40*	0.533	0.633	18.80
At 20 docs	0.317	0.167	-47.40	0.333	0.417	25.00	0.508	0.600	18.00
At 30 docs	0.339	0.183	-45.90	0.317	0.356	12.30	0.406	0.494	21.90
Avg Prec	0.147	0.088	-14.97	0.237	0.253	6.70	0.291	0.338	16.35

Topic	431i			438i			446i		
	Initial	Opt.	Diff (%)	Initial	Opt.	Diff (%)	Initial	Opt.	Diff (%)
No. of session	6	6		6	6		6	6	
Retrieved Rel	535	778	45.42	540	691	27.96	362	459	26.8
Precision									
At 5 docs	0.733	0.733	0.00	0.400	0.567	41.70	0.600	0.633	5.60
At 10 docs	0.717	0.683	-4.70	0.250	0.517	106.70	0.550	0.533	-3.00
At 20 docs	0.608	0.617	1.40	0.233	0.450	92.90	0.450	0.492	9.30
At 30 docs	0.550	0.533	-3.00	0.217	0.400	84.60	0.367	0.422	15.20
Avg Prec	0.338	0.418	23.59	0.109	0.182	66.32*	0.107	0.129	20.45

*** indicates the statistical significance at $p < 0.05$.**

strongest trend of the improvements in the menu groups was found in the precision at the document level of 1 to 30 (in Table 5) while the Baseline group was likely to improve at the lower document levels.

This suggests two points. One is that the optimised queries generated by the menu groups could be based on the selection of the relevant documents from a wider range of rankings than the Baseline. Another possibility is that such optimised queries should stand a better chance to bring up the rankings of a wider range of relevant documents.

3.3 User Perception

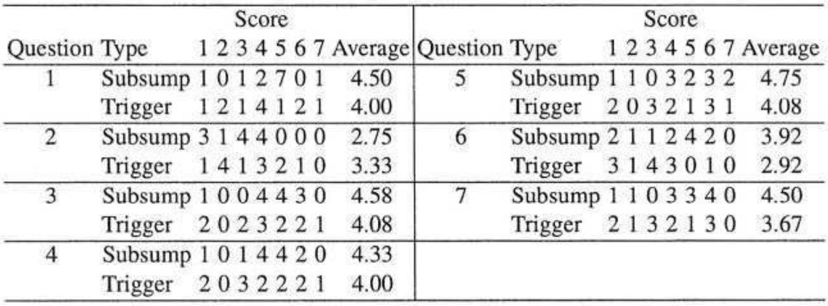
Now that the results based on the recall/precision and log analysis have been discussed, following two sections will present the results from the questionnaires and manual observations.

Subjects were asked to fill in a short questionnaire after each session. The following aspects of the CiQuest system were investigated by the questionnaire:

1. Ease of use of the system
2. Size of menus (Too long or too many?)
3. The menus as a tool to help predicting the contents of linked documents
4. The menus as a tool to help relevance judgement of documents
5. The menus as a tool to help focusing on important terms
6. The menus as a tool to help understanding the contents of documents
7. The menus as a tool to help having a better idea of a set of retrieved documents
8. Preference of system settings

The result of Question 1 to 7 is shown in Table 7.

Table 7. User perception (Score 1: Not at all, 4: Sometimes, 7: Always)



		Score										Score							
Question	Type	1	2	3	4	5	6	7	Average	Question	Type	1	2	3	4	5	6	7	Average
1	Subsump	1	0	1	2	7	0	1	4.50	5	Subsump	1	1	0	3	2	3	2	4.75
	Trigger	1	2	1	4	1	2	1	4.00		Trigger	2	0	3	2	1	3	1	4.08
2	Subsump	3	1	4	4	0	0	0	2.75	6	Subsump	2	1	1	2	4	2	0	3.92
	Trigger	1	4	1	3	2	1	0	3.33		Trigger	3	1	4	3	0	1	0	2.92
3	Subsump	1	0	0	4	4	3	0	4.58	7	Subsump	1	1	0	3	3	4	0	4.50
	Trigger	2	0	2	3	2	2	1	4.08		Trigger	2	1	3	2	1	3	0	3.67
4	Subsump	1	0	1	4	4	2	0	4.33										
	Trigger	2	0	3	2	2	2	1	4.00										

Use of system. Question 1 asked the subjects how easy it was to use the system, rated between 1 (Not at all) and 7 (Always). The table shows that the Trigger menu's score is distributed across the scale, whereas the majority scored the Subsump menu at 5.

Size of menu. Question 2 sought to establish to what extent the menus were considered to be too long or containing too many terms. The lower score is better in this question. The Subsump menu's score concentrated at the lower end of scales while the Trigger menu's ratings were distributed more widely. Nevertheless the size of the menus did not seem to overwhelm the subjects in either case.

Predicting contents. Question 3 asked how useful a menu was to predict the contents of documents linked to the terms in a menu. The menu was designed to show a set of documents linked to each term in the menu when a user clicked it. As can be seen, the majority of subjects (11) gave a score between 4 and 7 for the Subsump menu. Although there were fewer subjects (8) for the Trigger menu who gave a score in this range, it appears that both types of menus succeeded in predicting the contents of linked documents.

Relevance judgement. Question 4 asked how useful a menu was for judging the relevance of documents during the sessions. Although the instance finding task was not to find a relevant document, the task latently involved the assessment of relevance (i.e. no instance would be found in a non-relevant document). The table shows that more subjects with the Subsump menu gave a score between 4 and 7 than with the Trigger menu.

Focusing on important terms. Question 5 asked how useful a menus was for focusing on terms of interest. As described before, the hierarchy provided a means of narrowing down to a subset of retrieved documents regardless of its ranked position. The scores of both types of menus were well distributed in the range above 4. The Subsump menu seemed to gain a slightly higher overall score than the Trigger menu.

Understanding contents. Question 6 asked how useful a menu was to understand the contents of documents. The table shows that the scores for the Subsump menu are generally high with the score 5 as the peak while the Trigger menu has the peak at the score 3.

Better idea of retrieved documents. Question 7 asked if a menu provided a better idea of a set of retrieved documents as opposed to individual documents. Similar to the previous question the Subsump menu seemed to gain a higher overall score than the Trigger menu.

Preference of system setting. After the completion of all sessions the subjects were asked their preference among the three settings with the overall feedback against the system. Two points became clear from the final questionnaire. First, more than half of the subjects showed their preference for the Baseline system because of its simplicity and familiarity. Second, most subjects except two did not clearly notice the difference between the two types of menus in terms of how to organise terms. This point will be discussed further in a later section.

Summary. The subjective evaluation of the hierarchies was presented through the questionnaires. Generally the subjects find the Subsump menu more useful than the Trigger menu in supporting information access. The scores of the Trigger menu tend to be distributed across the scale, while for the Subsump menu they are concentrated at a higher level. A more detail comparison of the concepts generated in the two hierarchies should be carried out to gain a better insight of how users interpret those concepts.

3.4 Other User Behaviour

In addition to the precision/recall evaluation, analysis of system logs, and questionnaires, observations were made and recorded manually during the sessions, the following describe some typical user behaviours.

Accessing the hierarchies. The most commong approach for accessing the hierarchy was:

1. Submit a query;
2. Examine several records in the first page of the results; then
3. Browse the hierarchy.

This route seems to show that the primary concern in the search process is on the documents. However it was found that many subjects decided to browse the menus after the first-page examination, as opposed to going on to the next page. This also seems to be influenced by the results of the first-page examination. When a subject found a reasonable amount of relevant documents in the first page, they tended to go to the next page. The hierarchy seemed to be accessed more frequently when the subjects were less satisfied with the first page.

Using the hierarchies. It was observed that there were two typical ways of using the hierarchy. One was to focus on a subset of documents. This was the most popular way to use the menus as described above. However, another way was to assess the potential usefulness of terms. In other words, some subjects selected a term, examined the title and best paragraph of the top linked documents, selected another term, examined the list, and repeated this process.

Browsing the hierarchies. The top level terms of the menu seem to be very important for the subjects in using the hierarchy. In particular it was observed that the absence of query terms at the top level seemed to discourage browsing through the hierarchy. This happened more often in the Trigger menu than in the Subsump menu. Hence, the top level terms were regarded as a starting point.

Another observation is that the subjects tended to go back to the same parent term when one of its children was found to be useful, and try another child term.

A final comment is that users' browsing action (i.e. movement from one concept to another) tended to be carried out easily and speedly. Although the subjects commented that they were aware of some irrelevant concepts included in the hierarchies, they seemed to be capable of filtering out those concepts during their tasks.

4 Conclusion and Future Work

4.1 Conclusive Discussion

We presented a user study to investigate the usability of the CiQuest system that was designed to support interactive searches. Our focus was on the task-based evaluation of the system as well as the standard precision/recall measures. From the instance finding task, it was found that the Baseline was also effective due to the good performance of our IR system, but the precision can be improved with the hierarchies. The query optimising task indicated that the hierarchies could help improve the precision at the higher document levels (i.e. 5 to 30) more significantly than the Baseline.

Questionnaires and manual observation revealed that the hierarchical structures can be easily used and be useful to support the information accessing process. Also several interesting user behaviours that can be characteristic in the use of concept hierarchies were identified and discussed. The main highlights of our findings are:

1. the hierarchy is often accessed after an examination of the first page of search results;
2. accessing the hierarchies reduces the number of iterations and paging actions;
3. accessing the hierarchies increases the chance of finding relevant items more accurately than the Baseline;
4. the hierarchical structure helps the users to handle a large number of concepts; and finally,
5. subjects were not aware of the difference between two types of hierarchies.

4.2 Future Work

In the query optimising task the Trigger hierarchy seems to slightly outperform the Subsump hierarchy. However the questionnaire indicated that the Subsump hierarchy was preferable. This suggests both approaches have can be beneficial as a means of generating a concept hierarchy to support information retrieval. Therefore an integration of two approaches is worthy a further investigation.

Exploring other techniques to determine the hierarchical relations between concepts should also be examined. For example, we came across the research by Bookstein [15] during the development of our system. Their analysis of symmetric and asymmetric relations between terms by measuring clumping strength could also be of interest.

Acknowledgements. The authors thank to the participants of our experiments from Department of Information Studies and Computer Science at the University of Sheffield. This work was carried out as a part of the CiQuest project, which was funded by the Library and Information Commission (now *re*:source, the Council for Museums, Archives and Libraries). Any opinions, findings, and conclusions described here are the authors and do not necessarily reflect those of the sponsor.

References

1. Belkin, N.J., Cool, C., Head, J., Jeng, J., Kelly, D., Lin, S., Lobash, L., Park, S.Y., Savage-Knepshield, P., Sikora, C.: Relevance feeback *versus* local context analysis as term suggestion devices: Rutgers' trec-8 interactive track experience. In Voorheer, E.M., Harman, D.K., eds.: Proceedings of the 8th Text REtrieval Conference (TREC-8), Gaithersburg, MD, NIST (2000)
2. Ruthven, I.: Re-examining the potential effectiveness of interactive query expansion. In Callan, J., Cormack, G., Clarke, C., Hawking, D., Smeaton, A., eds.: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tronto, Canada, ACM (2003) 213–220
3. Pollitt, S.: Interactive information retrieval based on faceted classification using views. In: Proceedings of the 6th International Study Conference on Classification (FID/CR), London, UK, University College of London (1997) Available from `http://scom.hud.ac.uk/external/research/CeDAR/dorking.htm` [Accessed: 08/01/2004].
4. Joho, H., Coverson, C., Sanderson, M., Beaulieu, M.: Hierarchical presentation of expansion terms. In: Proceedings of the 17th ACM Symposium on Applied Computing (SAC'02), Madrid, Spain, ACM (2002) 645–649
5. Hersh, W., Over, P.: Trec-8 interactive track report. In Voorheer, E.M., Harman, D.K., eds.: NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC-8), Gaithersburg, ML, NIST (2000) 57–64
6. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In Hearst, M., Gey, G., Tong, R., eds.: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, ACM (1999) 206–213
7. Niwa, Y., Nishioka, S., Iwayama, M., Takano, A.: Topic graph generation for query navigation: Use of frequency classes for topic extraction. In: Proceedings of the Natural Language Processing Pacific Rim Symposium (NLPRS'97), Phuket, Thailand (1997) 95–100
8. Nanas, N., Uren, V., De Roeck, A.: Building and applying a concept hierarchy representation of a user profile. In Callan, J., Cormack, G., Clarke, C., Hawking, D., Smeaton, A., eds.: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tronto, Canada, ACM (2003) 198–204
9. Anick, P.G., Tipirneni, S.: The paraphrase search assistant: Terminological feedback for iterative information seeking. In Hearst, M., Gey, G., Tong, R., eds.: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, ACM (1999) 153–161
10. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the Fourteenth International Conference on Computational Linguistics, Nantes, France (1992)539–545
11. Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM **38** (1995) 39–41
12. Wacholder, N., Evans, D.K., Klavans, J.L.: Automatic identification and organization of index terms for interactive browsing. In: Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries, Roanoke, VA, ACM (2001) 126–134

13. Robertson, S., Walker, S., Hancock-Beaulieu, M.: Large test collection experiments on an operational, interactive system: Okapi at trec. Information Processing & Management **31** (1995) 345–360
14. Fowkes, H., Beaulieu, M.: Interactive searching behaviour: Okapi experiment for trec 8. In Robertson, S., Ayse, G., eds.: Proceedings of the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research, Cambridge, UK, BSC-IRSG (2000) 47–56
15. Bookstein, A., Kulyukin, V., Raita, T., Nicholson, J.: Adapting measures of clumping strength to assess term-term similarity. Journal of the American Society for Information Science and Technology **54** (2003) 611–620

Searcher's Assessments of Task Complexity for Web Searching

David J. Bell and Ian Ruthven[*]

Department of Computer and Information Sciences
University of Strathclyde, Glasgow, G1 1XH
{dbell,ir}@cis.strath.ac.uk

Abstract. The complexity of search tasks has been shown to be an important factor in searchers' ability to find relevant information and their satisfaction with the performance of search engines. In user evaluations of search engines an understanding of how task complexity affects search behaviour is important to properly understand the results of an evaluation. In this paper we examine the issue of search task complexity for the purposes of evaluation. In particular we concentrate on the searchers' ability to recognise the internal complexity of search tasks, how complexity is affected by task design, and how complexity affects the success of searching.

1 Introduction

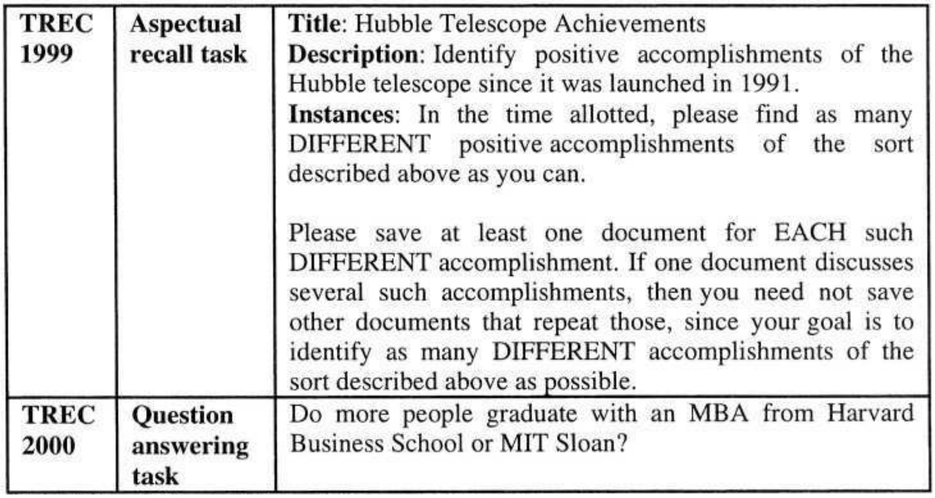
User evaluations of search systems and interfaces attempt to assess the utility of search tools when used by human searchers. One of the main components in such evaluations are search tasks; descriptions of an information need that can used by searchers to formulate search statements and assess the relevance of retrieved documents.

In operational evaluations, such as the ones described in [1], the search tasks come from the searchers themselves. These search tasks are ones that the searcher has encountered independently of the evaluation and reflect a searcher's personal information need. This type of search task provides realistic search scenarios with which to assess a search system.

More commonly, search tasks are used within laboratory evaluations in which the experimental designer creates a number of search tasks for use within the experiment. This means that the same tasks can be used across a range of experimental subjects and systems thus allowing for a comparison of search success under different experimental conditions. A good example of created search tasks can be found within the interactive track of TREC [8]. Here the use of created search tasks allows cross-site evaluation of search systems. The nature of search tasks used in TREC changes from year to year to investigate different types of search tasks. Figure 1 gives examples of typical TREC search tasks.

[*] Corresponding author.

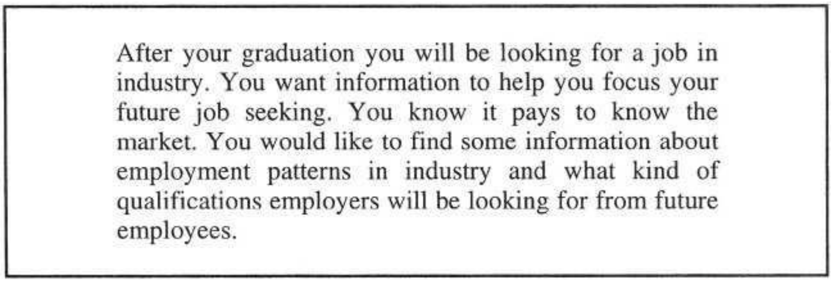
S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 57–71, 2004.



TREC 1999	**Aspectual recall task**	**Title**: Hubble Telescope Achievements **Description**: Identify positive accomplishments of the Hubble telescope since it was launched in 1991. **Instances**: In the time allotted, please find as many DIFFERENT positive accomplishments of the sort described above as you can. Please save at least one document for EACH such DIFFERENT accomplishment. If one document discusses several such accomplishments, then you need not save other documents that repeat those, since your goal is to identify as many DIFFERENT accomplishments of the sort described above as possible.
TREC 2000	**Question answering task**	Do more people graduate with an MBA from Harvard Business School or MIT Sloan?

Fig. 1. Example TREC Interactive Track topics

Borlund [3, 4] has promoted the use of simulated work task situations in order to create more realistic search tasks. Simulated work task situations are short search narratives that describe not only the need for information but also the situation – the work task – that led to the need for information. An example, taken from [4] is shown in Figure 2. Simulated work task situations are intended to provide searchers with a search context against which the searchers can make personal assessments of relevance.



After your graduation you will be looking for a job in industry. You want information to help you focus your future job seeking. You know it pays to know the market. You would like to find some information about employment patterns in industry and what kind of qualifications employers will be looking for from future employees.

Fig. 2. Example simulated work task situation from [4]

One important aspect in the creation of search tasks, whether through TREC-style topics or simulated work task situations, is how difficult it is to search using the task. Many factors can affect the difficulty of a search tasks, for example:

- *the difficulty of understanding what information is required.* Search tasks may require specialist knowledge about the task domain before starting a search, or the tasks may be too vaguely specified to allow the searcher to proceed with the search.
- *the difficulty of searching.* For some tasks it may be difficult to specify a search statement, or query, to submit to the retrieval system. For other tasks it may be

difficult to find information because the collection contains little information on a given topic.

- *the difficulty of interpreting relevance.* Depending on the searcher's knowledge of a topic, or previous searching experience, it may be difficult for a searcher to decide when a document contains relevant information. For example, in the question answering tasks in Figure 1, it may be easy to assert a document contains *an* answer but not when it contains a *correct* answer. For other tasks, it may be more difficult to decide on whether a document is relevant without more information on the task area or the context of the search.

These three areas affect different parts of a search; the initial pre-search understanding of a search task, the conversion of this conceptual understanding to a search statement, and the process of assessing retrieved material. Task difficulty therefore affects the whole search process and consequently our evaluation of search systems. Tasks that are too easy may result in too little interaction for analysis; tasks that are too difficult could result in low user commitment to the experiment. It is important therefore to be able to distinguish tasks according to how difficult they may prove in an experimental setting. In this paper we explore the nature of task difficulty, in particular the nature of *task complexity,* where task complexity is a measure of the uncertainty within a search task. We carry out a study on search tasks of varying complexity within a web search environment to investigate searchers' perceptions of task complexity and how these perceptions relate to characteristics of the search tasks.

We present our methodology and components of the study in section 3, the main findings in section 4, and discuss the limitations and implications in section 5. Prior to this, in section 2, we present an overview of task complexity for information seeking.

2 Related Work

The notion of a *task* in information seeking covers a range of interrelated concepts. For example, the *work* task, e.g. preparing a research paper, relates to the activity that results in a need for information, [2, 9]. A work task may give rise to several *search* tasks, the specific search on which a user is engaged. Each individual search task involves a series of tasks and decisions relating to operating the system and assessing search results [5].

Several studies on the impact of tasks on information seeking have pointed to the importance of task complexity and the variables that can affect complexity. Kelliher, for example, relates complexity to the number of decisions to be made and indicates that, when faced with highly complex tasks, decision-makers attempt to reduce complexity by eliminating alternative actions or outcomes [10]. Vakkari [11] surveys task complexity as it has been investigated within information seeking and relates the notion of task complexity to important variables such as prior searcher knowledge, search strategies and relevance. He points to the fact that although we can categorise some of the factors that affect complexity, task complexity is not an objective measure: personal factors can affect an individual's assessment of the complexity of a task.

Both Campbell [7] and Byström and Järvelin [6] have examined the factors that can make a task more or less complex. Campbell describes task complexity as a function of the psychological states of the task performer, the interaction between the task characteristics and the abilities of the task performer and the objective attributes of the task itself, such as the number of sub-tasks or the uncertainty of the task outcome. He proposes four attributes that can increase the complexity of a given task: multiple potential paths to a desired end-result, the presence of multiple desired outcomes, the presence of conflicting interdependencies between paths, and uncertainty regarding paths. These all can apply to information retrieval interaction; a searcher may obtain relevant information using different queries or search strategies *(multiple paths),* may require different pieces of information *(multiple outcomes),* paths may conflict (a searcher may have to split search tasks), and paths may be uncertain (the use of relevance feedback, for example, may have an unknown effect on the search). Based on the combination of these four attributes, Campbell proposed a categorization of 16 task types, e.g. simple tasks which contain none of the complexity-increasing attributes, and fuzzy tasks which contain both multiple end-states and multiple paths.

Byström and Järvelin also proposed a categorization of tasks, specifically related to information seeking and based on real-life information seeking situations [6]. This categorisation defines five levels of task complexity based on the *a priori determinability* of tasks. The *a priori determinability* is a measure of the extent to which the searcher can deduce the required task inputs (what information is necessary for searching), processes (how to find the required information) and outcomes (how to recognise the required information) based on the initial task statement. Increasing complexity is associated with increasing uncertainty regarding these factors, i.e. the less sure a searcher is about task inputs, search process or search outcomes, the more complex is the search task.

Byström and Järvelin's work was based on investigating real search behaviour in real work situations. As such it is wide in scope, incorporating aspects of the real work tasks as well as search tasks. The measure of complexity proposed in their study was based on retrospective analyses of the factors that increase or reduce the complexity of an information-seeking task.

In this study we use similar factors to test whether we can *predictively* influence the complexity of *artificial* search tasks; ones that may be applied to laboratory investigations. We also investigate how task complexity influences searchers' perceptions and satisfaction with the search process. As we discuss in section 5 the ability to manipulate and assess the complexity of search tasks can aid in the understanding and design of user evaluations.

3 Methodology of Study

In this study we create search tasks of varying complexity and use the tasks to analyse searchers' reactions to tasks of varying complexity. We use the search tasks within a laboratory evaluation methodology, similar to those used in evaluations such as TREC, to compare the complexity of tasks within the environment in which they would typically be used. In this section we describe the main components of the study: the creation of the search tasks (section 3.1), the search systems used (section 3.2) and the participants (section 3.3). In section 3.4 we describe the methodology itself.

3.1 Search Tasks

Our model of task complexity is based on the classification proposed by Byström and Järvelin [6]. They define a five-level categorization of task complexity. We conflate this model into a three-level model to create a better separation between the complexity of tasks.

- **Complexity level 1**[1] are tasks where the tasks are almost completely *a priori* determinable. It is generally clear what information is required, how to find the information and how to assess relevance. However, some parts of the search process or information needed may be vague.
- **Complexity level 2**[2] are tasks in which the desired information may be clear, however the searcher must make case-by-case decisions regarding the inputs and search process.
- **Complexity level 3**[3] are the most complex tasks. In this type of task the whole search may be unclear from the start, i.e. it is unclear what information is being sought, how to obtain relevant information and how the searcher will know they have found relevant information.

In the study we created three groups of search task. Each of these task groups contains three variations of an individual search task, each variation reflecting a different level of complexity. An example is shown in Figure 3 for task group C. In this case, each of the three task variations is centred around the same information need – *information on changes to petrol prices.* Increasing task complexity is associated with manipulating the factors that affect the *a priori* determinability factors related to the tasks. The first of these factors involves the information input to the task. This was altered by changing the amount of information, provided by the task description, that the participant will be able to use within the search. Task C1, for example, restricts the search to the *price of petrol in the UK in recent years,* the inputs *price, recent* and *UK* provide information that the searcher can use to understand what information is being sought. Task C3 on the other hand, provides fewer clues about information can be used to search.

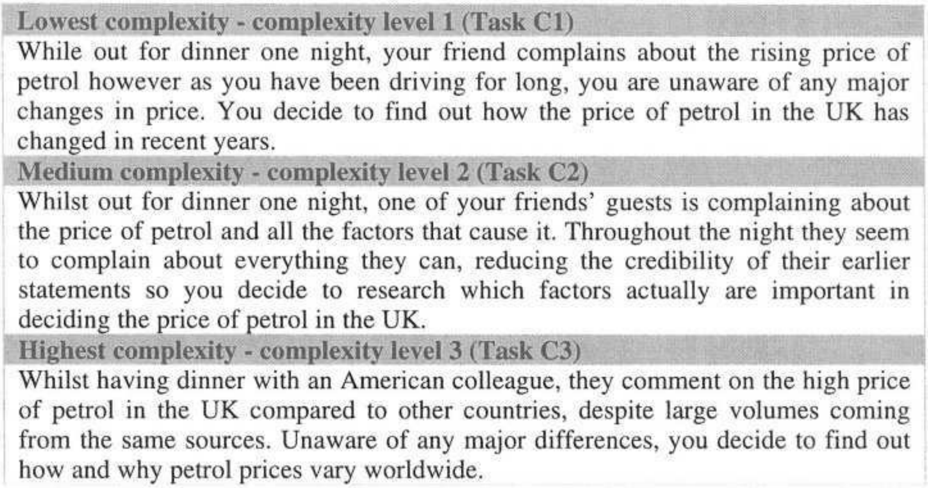
The second factor involves manipulating the process involved in finding the relevant information. A more complex task may involve comparing or analysing data from multiple sources. For the task group shown in Figure 3, the least complex task involves finding data related to petrol prices within the UK, the most complex task involves finding data related to worldwide prices. The most complex task, therefore, may not be answered by a single source, and the process of finding information becomes less clear from the start.

The final factor relates to the requested information output of the task – what information is required to complete the search task. This can be manipulated in two ways, by the amount of data required and the type of data required. For the tasks in

[1] Corresponds to the range of tasks between Byström and Järvelin's Automatic Information Processing Tasks and Normal Information Processing Tasks.

[2] Corresponds to Normal Decision Tasks.

[3] Corresponds to the range of tasks between Known Genuine Decision Tasks and – Genuine Decision Tasks.



Lowest complexity - complexity level 1 (Task C1)

While out for dinner one night, your friend complains about the rising price of petrol however as you have been driving for long, you are unaware of any major changes in price. You decide to find out how the price of petrol in the UK has changed in recent years.

Medium complexity - complexity level 2 (Task C2)

Whilst out for dinner one night, one of your friends' guests is complaining about the price of petrol and all the factors that cause it. Throughout the night they seem to complain about everything they can, reducing the credibility of their earlier statements so you decide to research which factors actually are important in deciding the price of petrol in the UK.

Highest complexity - complexity level 3 (Task C3)

Whilst having dinner with an American colleague, they comment on the high price of petrol in the UK compared to other countries, despite large volumes coming from the same sources. Unaware of any major differences, you decide to find out how and why petrol prices vary worldwide.

Fig. 3. Task group C

Figure 3, the least complex tasks limits the amount of data applicable (by requesting only recent information), the UK restriction means relevant data will likely only refer to certain units (currency and volumes) that are applicable to UK petrol prices. In contrast the most complex task asks for worldwide factors that influence prices increasing the amount of data that is applicable, and, as different factors may be important in different countries, increasing the type of factors that are applicable.

The investigation therefore contrasts increasing complexity across versions of the same search task. An alternative would have been to create unique tasks of varying complexity. However it can be difficult to assess the relative complexity of tasks on different topics. Our methodology reduces the overall number of search topics to be created, and allows comparison between different versions of the same core task. The tasks were framed within simulated work task situations to encourage personalised searching by the participants.

In pilot testing we created several task groups. The three search groups that displayed the best variation in task complexity, as assessed by participants in the pilot study, were chosen for the final study. The three search tasks will be referred to as groups A-C[4], within each group the individual search tasks are numbered from 1-3 with 1 reflecting the task with the lowest complexity, e.g. A1 is the task in group A with the lowest complexity.

3.2 Search Systems

In the study we asked the participants to search using the search tasks. We used two search interfaces. Both systems were interfaces to the WiseNut[5] internet search engine. Two search interfaces were employed in the study to be able to generalise searchers' assessment of search task complexity beyond the interface itself, i.e. so that

[4] Task groups A and B are given in the Appendix.

[5] http://www.wisenut.com/

the measurement of complexity is not solely a factor of the individual interface used. The interfaces are described in detail in [12, 13][6] in this section we shall only describe the main features of the two interfaces used. Screen-shots of the two interfaces are given in Appendix A.

The first interface, **Sum-Int,** is a summarisation interface [12], Appendix Figure A.1. Titles of retrieved web pages are shown in groups of ten and moving the mouse over the title of a retrieved page will displayed a short summary of the web page to the searcher. The summaries themselves are composed of the top four sentences in the web page that are the best match to the searcher's query.

The second interface, **TRS-Int,** also offers a summary of retrieved documents, Appendix Figure A.2. This interface also displays to the searcher a list of sentences taken from the top 30 retrieved documents, the *top-ranking sentences,* ranked in order of how well the sentence matches the searcher's query. The intention behind this feature is to help the searcher locate relevant information regardless of which document contains the information. This has previously been shown to be useful in helping the searcher identify relevant material [13]. In TRS-Int, each the title of each retrieved page is associated with a check-box. By clicking on the check-box the searcher can indicate to the system that the retrieved page contains useful information. If the searcher clicks on a check-box the contents of the page's summary is used to modify the searcher's query and the list of top-ranking sentences is updated to reflect the new query. This form of relevance feedback is intended to keep the most useful sentences at the top of the list of sentences.

3.3 Participants and Methodology

30 people participated in the main study: 9 female and 21 male. All participants were aged between 18 and 25 years and were university students from a variety of academic disciplines. Each participant was asked to search on three search tasks, one from each of the three search groups (A-C) and were given 5 minutes to search on each task. The time restriction was based on pilot testing which indicated that 5 minutes was sufficient time for most participants to complete most of the tasks. In presenting the tasks to the participants the order of search task *topic* was held constant (the participants received a task from group A followed by one from group B, finally a task from group C), however the complexity of the search tasks were rotated using a Greco-Latin square design, e.g. participant 1 received tasks A1, B2, C3, participant 2 received tasks A2, B3, C1, etc. None of the participants had previously used either search interface. Each participant searched only on one of the search interfaces to avoid the participants having to cope with two novel search interfaces.

4 Results

In this section we present the main results of this investigation. Our analysis is focussed on the three main aspects of the investigation: the participants' ability to

[6] We gratefully acknowledge the support of Ryen White of the University of Glasgow in providing these interfaces.

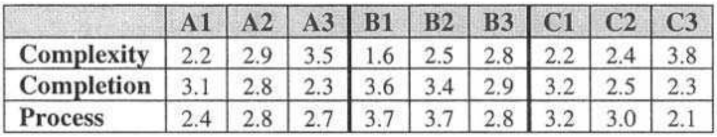
recognise task complexity, the factors that affect complexity and the relationship between complexity and the participants' interest in the tasks. In each of the following sections we will develop the main research hypotheses being investigated.

4.1 Participants' Perceptions of Complexity

In this section we investigate our core hypothesis, namely that by modifying the search tasks in the manner described in section 3.1 we create search tasks that have recognisably different levels of complexity. In one sense, this is a validation test for our approach to manipulating task complexity: if there is no difference between reported assessments of task complexity then it may that searchers *can* recognise task complexity but our method of creating complex tasks is poor. One the other hand, if the participants report clear differences in task complexity then we can conclude that task complexity can be recognised and that our method creates tasks of varying complexity. Our research hypothesis is, therefore, that participants can differentiate the complexity of the employed search tasks.

To investigate this, after each search, participants were asked to record the overall complexity of the search task on a 5-point scale in which a value of 1 reflected a task with little complexity and a value of 5 indicated a highly complex task. Table 1 (row 2) summarises the results from the participants' assessments of the tasks' complexity. As can be seen, for all task groups, the participants' rating of task complexity increases according to the predicted complexity of the task. This provides an initial validation of the method of varying task complexity. The responses for the tasks A1, and C3 were significantly different[7] from the other tasks in the task group and all tasks in group B were significantly different from each other.

Table 1. Average rating of task complexity



	A1	A2	A3	B1	B2	B3	C1	C2	C3
Complexity	2.2	2.9	3.5	1.6	2.5	2.8	2.2	2.4	3.8
Completion	3.1	2.8	2.3	3.6	3.4	2.9	3.2	2.5	2.3
Process	2.4	2.8	2.7	3.7	3.7	2.8	3.2	3.0	2.1

Following from this initial hypothesis we investigate two possible related aspects; perceived task completion Table 1 (row 3) and ease of finding information Table 1 (row 4). In particular we measured responses to the degree to which the participants felt they had completed the task and how simple they felt it to find information (process). Both are measured on a 5-point scale in which a value of 5 reflects greater sense of task completion or a simpler process of finding information.

Generally the participants' assessment of task completion was inversely correlated with their assessments of task complexity; the more complex a task was rated, the less likely the participants were to feel that they had completed the task.[8] The actual correlation figures are discussed in section 4.4.

[7] Using a one-tailed Mann-Whitney test for independent samples, $p<0.05$

[8] Significance testing showed significant differences between the scores for tasks A1/A2, A1/A3, B1/B3 and C1/C3.

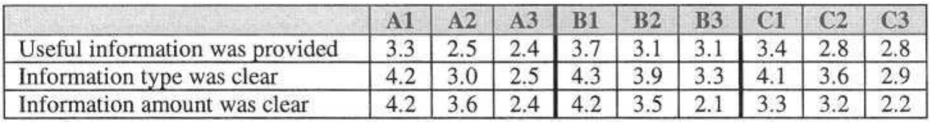
For task groups B and C there was also an inverse correlation with the participants' assessments of how simple was the process of finding information: the more difficult was the process of finding information the more complex the task was perceived as being.[9] However, this is does not hold for task group A. There is, therefore, some support for the difficulty of finding information, while not a complete determinant in the assessment of complexity, playing a part in complexity. In the next section we examine what causes the difference in complexity assessment.

4.2 Factors Affecting Complexity

The factors that were used to differentiate between the tasks in each task group were related to the *a priori* determinability of the search task; based on the task description how easy was it for the searcher to elicit useful information from the task description on what information was required, how easy was it to recognise relevant information and how clear was it to decide how relevant information was to be found.

To investigate which of these factors affected complexity, the participants were asked to rate the tasks according to three questions, again using a 5-point scale with 5 reflecting highest level of agreement: *'Useful information was provided by the task', 'The type of information to be retrieved was clear'* and *'The amount of information to be retrieved was clear'.* Table 2 summarises the participants' responses. Generally we would predict that the values would decrease from left to right, i.e. as less useful information is provided, or less information on the type or amount of information required is given, then task complexity would increase. Even though the differences between the scores for utility of information were slight, this relationship generally holds across the tasks with the higher complex tasks receiving scores less than or equal to the less complex tasks.[10] Therefore as the task expresses less useful information on what information is required, or less information on the type or amount of information to be retrieved, the participants perceive the task to be more complex.

Table 2. Participant responses to complexity increasing factors



	A1	A2	A3	B1	B2	B3	C1	C2	C3
Useful information was provided	3.3	2.5	2.4	3.7	3.1	3.1	3.4	2.8	2.8
Information type was clear	4.2	3.0	2.5	4.3	3.9	3.3	4.1	3.6	2.9
Information amount was clear	4.2	3.6	2.4	4.2	3.5	2.1	3.3	3.2	2.2

4.3 Personal Reactions to the Search Tasks

As mentioned previously, a searcher's estimate of task complexity can be influenced by subjective factors such as how much knowledge the searcher has about the task. In

[9] Significance testing showed significant differences between the scores for tasks B1/B3, B2/B3, C1/C3 and C2/C3.

[10] Significant differences between comparisons A1/A2, A1/3, C1/C2 for utility of information, A1/A2, A1/A3, B1/B3, B2/B3, C1/C3 for type of information required, and A1/A2, A1/3, B1/B2, B1/B3, B2/B3, C1/C3, C2/C3 for amount of information required.

this section we examine the participants' reactions to the assertions *'This task was easy to understand', 'The task was interesting'* and *'The task was relevant to me'*. In Table 3 we summarise the participants' responses. Answers are given on a 5-point scale, with a value of 5 reflecting the highest level of agreement.

Table 3. Participant responses to personal reactions

	A1	A2	A3	B1	B2	B3	C1	C2	C3
Easy to understand	3.5	2.9	3.2	4.1	3.8	3.1	3.8	3.1	3.0
Task was interesting	3.1	2.7	2.9	3.9	3.5	3.3	2.8	2.3	3.0
Task was relevant to me	3.6	3.4	3.5	3.8	3.2	3.1	3.1	2.5	2.7

There were few patterns regarding the latter two aspects (task interest and task relevance) except that tasks that had a lower complexity were more likely to be judged as more interesting or relevant than more complex tasks. However, there were no significant differences found regarding these two aspects. Within each group the search tasks were based on the same core topic, e.g. changes in petrol prices, therefore we might not expect any differences between the responses *within* a task group. That is, we might not expect a participant to be more *interested* in the topic of petrol prices whether the topic is placed within a highly complex or less complex search task. The lack of significant differences across task groups might simply reflect the individual differences in topic interest among our subjects. On the other hand, it may also reflect the fact that searchers who are closer to task completion, those who are searching on less complex tasks, are more likely to have obtained interesting information earlier in the search.

A similar pattern arises regarding how easy it was to understand the task with the lowest level complexity being rated as easier to understand on all groups. Here there were stronger differences, with, B3 significantly lower than B1, and C1 significantly higher than C2 or C3. So the ability to understand the task is related to the assessed complexity. From discussions with the participants, this was related to the *a priori* determinability: the participants' ability to understand what was required from reading the search task.

4.4 Correlation Analyses

In this section we examine the correlation of participants assessments of complexity against the various aspects described in sections 4.1 to 4.3 to compare the relative importance of each aspect. In Table 4 we show the results of applying Spearman's Rank Correlation Coefficient to the participants responses.

Table 4. Correlation of questionnaire responses with assessments of task complexity

	Process	Completion	Useful	Type	Amount	Understanding	Interesting	Relevant
A	-0.24	-0.22	-0.33	-0.65	-0.66	-0.12	0.00	0.34
B	-0.73	-0.59	-0.40	-0.68	-0.79	-0.71	-0.56	-0.31
C	-0.53	-0.53	-0.48	-0.74	-0.69	-0.49	0.05	-0.12

Across the task groups there was a constant relatively high inverse correlation of complexity with the type and amount of information required being clear from the task. Indeed, for each task group the strongest correlation was with the amount of information required. There was generally little correlation, however, with how interesting or relevant the task was to the searcher although the ability to understand the task set was important in task groups B and C. In this study, therefore, the information requirements of the task – how much information is required and what type of information – and the searcher's ability to understand these requirements appear to be more strongly linked to complexity than issues such as interest or relevance.

In task groups B and C the complexity was inversely related to task completion and the reported simplicity of the information-seeking process. This demonstrates the importance of assessing complexity when assessing the results of user evaluations.

4.5 Cross-System Comparisons

As mentioned in section 3.2 we used two search interfaces to be able to generalise the results beyond a single interface. We compared the results of the questionnaires for each task when performed on the two interfaces using a two-tailed Mann-Whitney test for independent samples, $p<0.05$. Although the numbers of responses used for each comparison are small, there were no significant differences found with the exception of responses to the assertion *'The task was relevant to me'* which were significantly higher for task C3 on the TRS-Int than on the Sum-Int.

5 Discussion

This study examined the impact of search task complexity on web searching. We created sets of search tasks using Byström and Järvelin's five-level model as the basis of our characterisation of task complexity. Using the created search tasks we examined whether web searchers could recognise task complexity and how this impacted on issues such as search success and searcher satisfaction.

There are several limitations to the study. For example, our study is limited in only examining search tasks rather than the whole work task that promotes individual search tasks. Also, our subjects only experienced one task from each complexity level rather than running several tasks from each level. Finally, the differentiation between the complexity of individual search tasks may not have been sufficiently great to properly determine the effect of complexity on other factors such as searcher understanding. Creating search tasks itself can be a difficult task as tasks can vary along other dimensions as well as complexity and these dimensions can interact. For example, one repeated comment was that some tasks were more complex because there was limited information available.

Our main aim is to promote task complexity as a factor in designing and interpreting user evaluations. In such evaluations, e.g. [12], questionnaire results on aspects such as searcher satisfaction, task completion, etc. are used in a comparative situation, e.g. System A leads to greater satisfaction than System B. However, it is not only the relative findings that are important; the absolute scores given to questionnaire responses are also important. If few searchers report reasonable search satisfaction, or task completion then the evaluation itself may be flawed. Assessing task complexity in pilot or pre-testing can be a useful method of determining whether search tasks are appropriate for individual evaluations. The method of using the same basic task, but varying the complexity, can elicit which version of a task is most appropriate for a given experimental study.

The *a priori* determinability can be used as a simple means of initially varying the tasks for pre-testing but the actual task complexity can be amplified or reduced by other factors such as the searcher's interest in the topic. This also provides additional support for Borlund's assertion that search tasks should be tailored to the experimental subject group [3].

Finally, we should recognise that task complexity is a dynamic entity. Tasks that offer little complexity may be answered quickly and by one document. However, tasks that are more complex may require several searches and aggregation of information from several documents or several search iterations. In user evaluations it is common to allow searchers only a certain time-frame in which to complete a search. This allows for stricter comparison between searches by different people or searches on different search systems. However, if the time given is too short then the searcher may not move from the stage of collecting information to the process of deciding on relevance and completing the search task. Therefore tasks may seem more complex at earlier search stages, when the searcher is collecting information, than in later search stages. In evaluations we should select appropriately complex tasks for the time we give to searchers or, alternatively, use measures of complexity to decide how much time we should allow searchers to complete a search task.

In summary, our findings validate Byström and Järvelin's model of task complexity and propose this model as a means of predicting and manipulating task complexity. The findings also indicate that task complexity should be seen as an integral part of designing and interpreting user evaluation results.

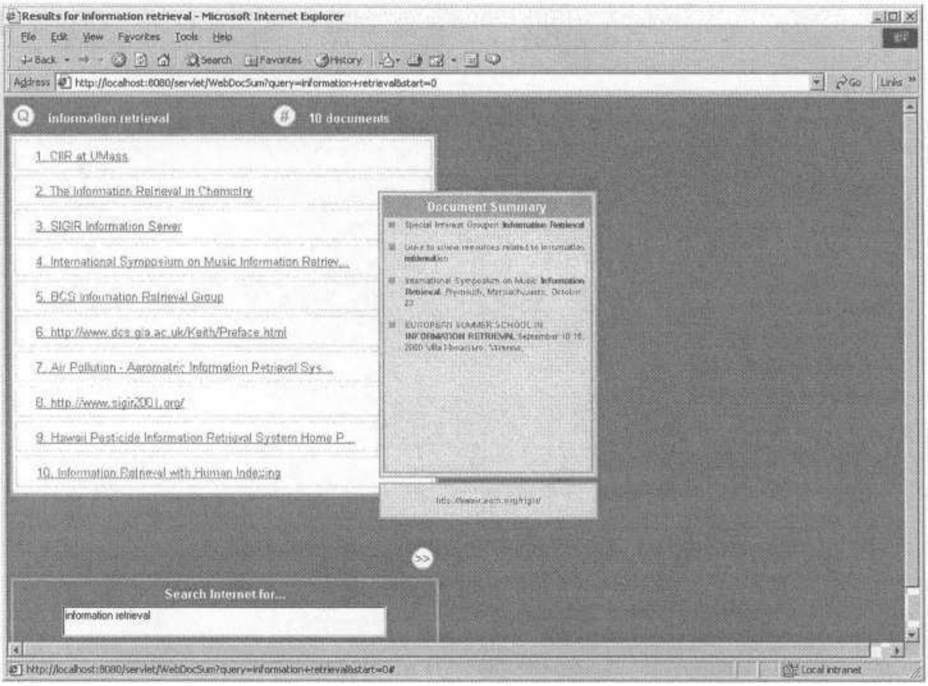
Acknowledgements. We would like to acknowledge the contribution of our participants to the work described in this paper and the many helpful comments from the anonymous reviewers.

References

1. M. Beaulieu. Experiments with interfaces to support query expansion. Journal of Documentation. 53. 1. pp 8-19. 1997.
2. N. J. Belkin, H. M. Brooks and R. N. Oddy. Ask for information retrieval. Journal of Documentation. 38. 2. pp 61-71. 1982.
3. P. Borlund. Experimental components for the evaluation of interactive information retrieval systems. Journal of Documentation. 56. 1. pp 71-90. 2000.
4. P. Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. Information Research. 8. 3. 2003.
5. M. K. Buckland, and D. Florian. Expertise, task complexity, and artificial intelligence: A conceptual framework. Journal of the American Society for Information Science. 42. 9. pp 635-643. 1991.
6. K. Byström and K. Järvelin. Task complexity affects information seeking and use. Information Processing and Management. 31. 2. pp 191-213. 1995.

7. D. Campbell. Task complexity: a review and analysis. Academy of Management Review. 13. pp 40-52. 1988.
8. W. Hersh and P. Over. The TREC 2001 interactive track NIST Special Publication 500-250: The Tenth Text Retrieval Conference (TREC 2001). p 38. 2002.
9. P. Ingwersen. Information retrieval interaction. Taylor Graham. London. 1992.
10. C. Kelliher. An empirical investigation of the effects of personality type and variation in information load on the information search strategies employed by decision-makers. Texas A&M University Ph.D. 1990.
11. P. Vakkari. Task complexity, information types, search strategies and relevance: integrating studies on information seeking and retrieval. Information Processing and Management. 35. 6. pp 819-837. 1999.
12. R. W. White, J. M. Jose and I. Ruthven. A task-oriented study on the influencing effects of query-biased summarisation in web searching. Information Processing and Management. 39. 5. pp 707-733. 2003.
13. R. White, I. Ruthven and J. Jose. Finding relevant documents using top-ranking sentences: an evaluation of two alternative schemes. Proceedings of the 25th Annual International ACM SIGIR Conference (SIGIR 2002). Tampere. pp 57-64. 2002.

Appendix A



Results for information retrieval - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go
Links
information retrieval
10 documents
Document Summary
>>
Search Internet for...
Local intranet

Address: http://localhost:8080/servlet/WebDocSum?query=information+retrieval&start=0

1. CIIR at UMass
2. The Information Retrieval in Chemistry
3. SIGIR Information Server
4. International Symposium on Music Information Retriev...
5. BCS Information Retrieval Group
6. http://www.dcs.gla.ac.uk/Keith/Preface.html
7. Air Pollution - Aeromatric Information Retrieval Sys...
8. http://www.sigir2001.org/
9. Hawaii Pesticide Information Retrieval System Home P...
10. Information Retrieval with Human Indexing

Document Summary

- Special Interest Group on **Information Retrieval**
- Links to other resources related to information **retrieval**
- International Symposium on Music **Information Retrieval**. Plymouth, Massachusetts, October 23
- EUROPEAN SUMMER SCHOOL IN **INFORMATION RETRIEVAL** September 10-15, 2000 Villa Monastero, Varenna,

http://www.acm.org/sigir/

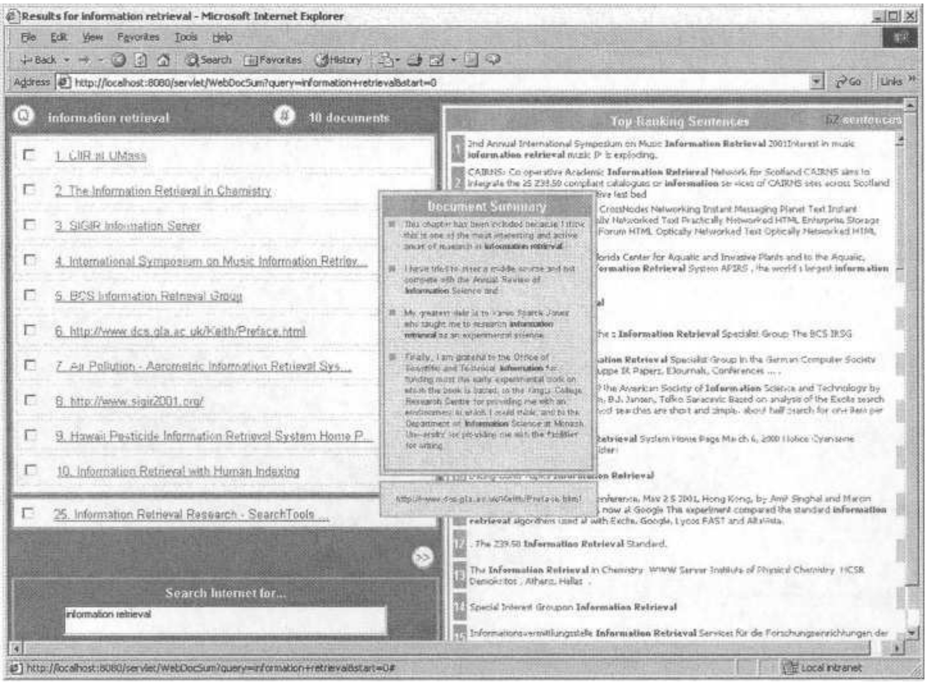
Search Internet for...: information retrieval

http://localhost:8080/servlet/WebDocSum?query=information+retrieval&start=0#

Fig. A.1. Interface one

Table A.1. Task groups A

Lowest complexity (Task A1)
A friend has recently been applying to various universities and courses but has been complaining that they are finding it difficult to attain a place due to the rising numbers of students. You were unsure if their assessment was correct so you have decided to find out how the size of the student population changed over the last 5 years and how it is expected to change over the coming 5 years.
Medium complexity (Task A2)
A friend has recently been applying to various universities and courses but has been complaining that they are finding it difficult to attain a place due as a much larger and varied number of people are attending university. You were unaware if their assessment was correct so you have decided to find out how the composition of the student population has changed over the last 5 years.
Highest complexity (Task A3)
A friend who has been attempting to gain a university place has been complaining that there are too many people attending university today, you were unsure if this assessment was correct and have decided to find out what changes there have been in the student population in recent times.



Results for information retrieval - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go
Links

http://localhost:8080/servlet/WebDocSum?query=information+retrieval&start=0

information retrieval 10 documents

- □ 1. CIIR at UMass
- □ 2. The Information Retrieval in Chemistry
- □ 3. SIGIR Information Server
- □ 4. International Symposium on Music Information Retriev...
- □ 5. BCS Information Retrieval Group
- □ 6. http://www.dcs.gla.ac.uk/Keith/Preface.html
- □ 7. Air Pollution - Aerometric Information Retrieval Sys...
- □ 8. http://www.sigir2001.org/
- □ 9. Hawaii Pesticide Information Retrieval System Home P...
- □ 10. Information Retrieval with Human Indexing
- □ 25. Information Retrieval Research - SearchTools ...

Search Internet for...

information retrieval

Document Summary

- This chapter has been included because I think this is one of the most interesting and active areas of research in **information retrieval**.
- I have tried to steer a middle course and not compete with the Annual Review of **Information** Science and
- My greatest debt is to Karen Sparck Jones who taught me to research **information retrieval** as an experimental science.
- Finally, I am grateful to the Office of Scientific and Technical **Information** for funding most the early experimental work on which the book is based, to the Kings College Research Centre for providing me with an environment in which I could think, and to the Department of **Information** Science at Monash University for providing me with the facilities for writing.

http://www.dcs.gla.ac.uk/Keith/Preface.html

Top Ranking Sentences 62 sentences

1 2nd Annual International Symposium on Music **Information Retrieval** 2001Interest in music **information retrieval** music IP is exploding.

2 CAIRNS: Co-operative Academic **Information Retrieval** Network for Scotland CAIRNS aims to integrate the 25 Z39.50 compliant catalogues or **information** services of CAIRNS sites across Scotland ...tive test bed

... CrossNodes Networking Instant Messaging Planet Text Instant ...lly Networked Text Practically Networked HTML Enterprise Storage ...Forum HTML Optically Networked Text Optically Networked HTML

...lorida Center for Aquatic and Invasive Plants and to the Aquatic, ...formation Retrieval System APIRS, the world's largest **information**

...**al**

...the : **Information Retrieval** Specialist Group The BCS IRSG

...**ation Retrieval** Specialist Group in the German Computer Society ...uppe IR Papers, EJournals, Conferences ... .

...f the American Society of **Information** Science and Technology by ...n, B.J. Jansen, Tefko Saracevic Based on analysis of the Excite search ...ted searches are short and simple, about half search for one item per

...**etrieval** System Home Page March 6, 2000 Notice Cyanamide ...lder

...**on Retrieval**

...onference, May 2-5 2001, Hong Kong, by Amit Singhal and Marcin ..., now at Google This experiment compared the standard **information retrieval** algorithms used at with Excite, Google, Lycos FAST and AltaVista.

12 . The Z39.50 **Information Retrieval** Standard.

13 The **Information Retrieval** in Chemistry WWW Server Institute of Physical Chemistry NCSR Demokritos , Athens, Hellas .

14 Special Interest Groupon **Information Retrieval**

15 Informationsvermittlungsstelle **Information Retrieval** Services für die Forschungseinrichtungen der

http://localhost:8080/servlet/WebDocSum?query=information+retrieval&start=0#

Fig. A.2. Interface two

Table A.2. Task groups B

Lowest complexity (Task B1)
Whilst in a mobile phone shop, you overhear a staff member telling one of their friends to wait until 3G or 3rd Generation phones are available before purchasing a new one. The staff are looking for a quick sale and don't seem to be very forthcoming with information on this technology so you decide to find out yourself what special features will be available on 3G or 3rd Generation mobile phones before making a decision.
Medium complexity (Task B2)
Whilst in a mobile phone shop, you overhear a staff member telling one of their friends to wait until 3rd Generation phones are available before purchasing a new one. The staff are looking for a quick sale and don't seem to be very forthcoming with information on this technology so you decide to find out yourself what special features will be available on 3rd Generation mobile phones before making a decision.
Highest complexity (Task B3)
Whilst in a mobile phone shop, you overhear a staff member telling one of their friends to wait to buy a 3rd Generation phone. Your friend didn't want to be sucked into buying something that may soon be obsolete so has asked you to explain 3rd Generation mobile phone technology to them.

Evaluating Passage Retrieval Approaches for Question Answering

Ian Roberts and Robert Gaizauskas

Department of Computer Science, University of Sheffield,
Regent Court, 211 Portobello Street, Sheffield, UK
{i.roberts,r.gaizauskas}@dcs.shef.ac.uk

Abstract. Automatic open domain question answering (QA) has been the focus of much recent research, stimulated by the introduction of a QA track in TREC in 1999. Many QA systems have been developed and most follow the same broad pattern of operation: first an information retrieval (IR) system, often passage-based, is used to find passages from a large document collection which are likely to contain answers, and then these passages are analysed in detail to extract answers from them. Most research to date has focused on this second stage, with relatively little detailed investigation into aspects of IR component performance which impact on overall QA system performance. In this paper, we (a) introduce two new measures, *coverage* and *answer redundancy,* which we believe capture aspects of IR performance specifically relevant to QA more appropriately than do the traditional recall and precision measures, and (b) demonstrate their use in evaluating a variety of passage retrieval approaches using questions from TREC-9 and TREC 2001.

1 Introduction

The question answering (QA) evaluations in the Text REtrieval Conferences of 1999–2003, have encouraged wide interest in the QA problem. A review of the proceedings papers for TREC 2001 (in particular [1]) shows that the majority of systems entered in this evaluation operate using a broadly similar architecture. First, an information retrieval (IR) system is used to retrieve those documents or passages from the full collection which are believed to contain an answer to the question. In many cases, the question words are simply used as-is to form the retrieval system query, though a few systems make use of more advanced query processing techniques. Then, these retrieved passages are subjected to more detailed analysis, which may involve pattern matching or linguistic processing, in order to extract an answer to the question. The main differences between the competing systems lie in the details of this second stage.

There are several reasons for this two-stage architecture. Foremost is the relative efficiency of IR systems in comparison with the more complex and unoptimized natural language processing (NLP) techniques used in answer extraction. Most answer extraction components of QA systems simply could not be run in any reasonable time over document collections of the size of the TREC

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 72–84, 2004.

collection. This is due in part to more extensive processing, which may include part-of-speech tagging, semantic tagging or shallow parsing, but also because few NLP researchers have spent time separating "index-time" from "search-time" functionality and devising data structures to optimize the latter (though see [2, 3] for exceptions). Another reason for separating retrieval and answer extraction is that IR researchers have spent decades designing systems to achieve the best possible performance, in terms of precision and recall, in returning relevant documents from large text collections. To most NLP researchers it has seemed self-evident that one should take advantage of this work.

Given this two-stage architecture, most of the attention of the QA community has focused on the answer extraction component of QA systems. The first stage IR component is simply treated as a black-box and relatively little work has been done to investigate in detail the effect that the quality of the IR stage has on systems' performance. Clearly, however, the second stage process can only determine an answer to a question if the passages retrieved by the first stage contain the necessary information. Furthermore, since, as Light et. al. [4] have shown, question answering systems tend to perform better for questions for which there are multiple answer occurrences in the document collection, an IR component that returns many occurrences of the answer in its top ranked documents is likely to be of more use in a QA system than one which returns few. QA systems such as the one developed by Microsoft Research [5] exploit this effect by searching for answers on the Web since, due to its huge size, it is likely that many more instances of the answer will be found than in the (relatively) small TREC collection.

In this paper, we concentrate on analyzing the performance of several different approaches to the information retrieval stage of QA, using measures which aim to capture aspects of performance relevant to question answering. More specifically, we concentrate on investigating several different approaches to passage retrieval (PR). For a typical TREC question, such as *Where is the Taj Mahal?,* only a small section of a document will be required to determine the answer. Indeed, supplying a QA system with the full text of the document may in fact be counter-productive, as there will be many more opportunities for the system to become distracted from the correct answer by surrounding "noise". Therefore using an IR stage which supplies the QA system with limited-length "best" passages is an approach which many QA researchers have adopted, and is the approach we investigate here. Given significant variation in document length across the TREC collection, passage retrieval approaches have the additional benefit of permitting processing-bound answer extraction components to examine passages further down the passage ranking than would be possible were full documents to be used.

Deciding to adopt a passage retrieval approach, as opposed to a document retrieval approach, is still indeterminate in several regards. Different approaches to passage retrieval assume different notions of passage. Well-known distinctions [6] are between semantic, discourse and window-based notions of passage, in which passage boundaries are seen as marked by topic shifts, discourse markers, such as paragraph indicators, or fixed byte spans, respectively. Furthermore, regardless of which notion of passage one adopts, a number of additional choices must be made in deciding how best to implement passage retrieval for QA. For instance, do we divide documents into passages prior to indexing, and make the passage the unit of retrieval, or dynamically at search time after ranking the document collection overall? These two approaches might lead to significantly different rankings of the same passages, and this difference could have important implications for QA.

In the following paper we investigate a number of different approaches to passage retrieval for QA using two new measures which we believe are more helpful in capturing aspects of IR system performance of relevance in the QA setting than the conventional measures of recall and precision. This work is by no means exhaustive in terms of the PR approaches considered, and does not aim to be. Its central contribution is to introduce measures by which one can assess passage retrieval for question answering and to initiate debate about which approaches to PR may be best for QA.

2 Measures for Evaluating IR Performance for QA

In the context of the QA task, the traditional IR performance measures of recall and precision demonstrate shortcomings that prompt us to define two new measures.

Let Q be the question set, D the document (or passage) collection, $A_{D,q}$ the subset of D which contains correct answers for $q \in Q$, and $R^S_{D,q,n}$ be the n top-ranked documents (or passages) in D retrieved by a retrieval system S given question q.

The *coverage* of a retrieval system S for a question set Q and document collection D at rank n is defined as

$$coverage^S(Q, D, n) \equiv \frac{|\{q \in Q | R^S_{D,q,n} \cap A_{D,q} \neq \emptyset\}|}{|Q|}. \quad (1)$$

The *answer redundancy* (or simply *redundancy*) of a retrieval system S for a question set Q and document collection D at rank n is defined as

$$redundancy^S(Q, D, n) \equiv \frac{\sum_{q \in Q} |R^S_{D,q,n} \cap A_{D,q}|}{|Q|}. \quad (2)$$

The coverage gives the proportion of the question set for which a correct answer can be found within the top n passages retrieved for each question. The answer redundancy gives the average number, per question, of passages within the top n ranks retrieved which contain a correct answer.

In this framework, *precision* is defined as

$$precision^S(Q, D, n) \equiv \frac{\sum_{q \in Q} \frac{|R^S_{D,q,n} \cap A_{D,q}|}{|R^S_{D,q,n}|}}{|Q|}, \quad (3)$$

and *recall* as

$$\mathit{recall}^S(Q,D,n) \equiv \frac{\sum_{q\in Q} \frac{|R^S_{D,q,n} \cap A_{D,q}|}{|A_{D,q}|}}{|Q|}, \tag{4}$$

with $\mathit{recall}^S(Q,D,n) = 0$ if $A_{D,q}$ is empty. The precision of a system for a given question set and document collection at rank n is the average proportion of the n returned documents or passages that contain a correct answer. Recall is the average proportion of answer bearing documents that are present in the top n returned documents or passages. In a QA context these global measures are not as helpful as coverage and redundancy. For example, suppose $n = 100$ and $|Q| = 100$. An IR system S_1 returning passages containing 100 correct answers in the top 100 ranks for a single question in $|Q|$ but 0 correct answers for all other questions receives the same precision score as a system S_2 returning exactly one correct answer bearing passage for each of the 100 questions in $|Q|$. However, S_1 when coupled to an answer extraction component of a QA system could answer at most one question correctly, while the S_2-based system could potentially answer all 100 questions correctly. Precision cannot capture this distinction, which is crucial for QA; coverage, on the other hand, captures exactly this distinction, in this case giving S_1 a score of 0.01 and S_2 a score of 1.

Recall is not as unhelpful as precision, and indeed one could argue that is more useful than redundancy as a measure, because it reveals to what extent the returned document set approaches the maximum redundancy obtainable, i.e. the extent to which all possible answering bearing passages are being returned. Redundancy, on the other hand, tells one only how many answering bearing passages per question are being returned on average. However, redundancy gives a neat measure of how many chances per question on average an answer extraction component has to find an answer, which is intuitively of interest in QA system development. More importantly, redundancy, being an absolute measure, can be compared across question and documents sets to give a measure of how difficult a specific QA task is. Furthermore, what answer redundancy misses, as compared to recall, can easily be captured by defining a notion of *actual redundancy* as $\sum_{q\in Q} |A_{D,q}|/|Q|$. This is the maximum answer redundancy any system could achieve. Comparing answer redundancy with actual redundancy captures the information that recall supplies, while giving overall information about the nature of the challenge presented by a specific question and document set which recall does not capture.

To obtain values for any of these measures, we must first decide what it means for an answer to be correct. In TREC, an answer is *correct* if it is a valid response to the question *and* if the document from which the answer is drawn provides evidence for the answer. This reflects the fact that an average user of a QA system does not trust the system absolutely, so an answer would only be accepted by the user if they could, in principle, verify it by reference to the original document. A candidate answer which is a valid response to the question, but which could not have been determined from the source document, is considered *unsupported.* Any other candidate answer is considered *incorrect.*

The judgment of an answer's correctness or otherwise is determined by a human assessor.

While this kind of manual evaluation is feasible for a one-off evaluation such as TREC, a similar process is not reasonable for repeated experiments on the retrieval system. An assessor would have to examine every passage retrieved to determine whether it (a) contained an answer to the question and (b) supported that answer. With potentially hundreds of passages to examine per question and hundreds of questions in the test set, this adds up to several hundred thousand passages per run. Also, since human judgments are inherently subjective, the same set of answers to the same questions, based on the same documents, will be scored differently by different assessors, so the results will not be repeatable. Clearly, an automatic method of assessment is needed.

Voorhees and Tice [7] describe a partial solution to this problem. For the TREC collection, NIST have created regular expression patterns, intended to match strings which answer each question, and a set of relevance judgments, assembled from the combined results of all participating systems, that indicate which documents provide supporting evidence for answers to each question. For our purposes, a passage is considered to contain a correct answer to a question if some substring of the passage matches one of the regular expressions for the question, and the document from which the passage was drawn is judged to be relevant.[1] The NIST automated approach to scoring QA systems is known to have limitations, as discussed by Voorhees and Tice; however, it is the only feasible approach for the sort of study we carry out here.

3 Alternative Approaches to Passage Retrieval

For the TREC-9 QA track our QA system [8], which adopts the two stage model for QA introduced in section 1, employed Okapi [9] as the IR component. For the reasons outlined in section 1 we wanted to use a passage-based approach and so relied upon Okapi's native support for paragraph-based passage retrieval. While using the native passage retrieval support of an IR engine such as Okapi was convenient, we became aware that the technique used by the engine might not be the most suitable for the question answering application. For example, Okapi will never retrieve more than one passage from the same source document, though it is quite possible that several such passages may be relevant to the question. There are essentially two ways to address this issue:

1. Pre-process the document collection, breaking documents into their component passages before indexing. The retrieval system then treats each passage as a document in its own right.

[1] If a question has multiple possible answers, it is possible that the passage contains one of these answers, but the document from which the passage was drawn supports a different answer, but we believe such situations to be sufficiently rare that they will not be considered further.

2. Retrieve full documents from the retrieval system, then break each document into its component passages and perform a second retrieval run to find the best passages across the retrieved document set.

With this context in mind, and keeping open the possibility that full document-based ranking may be superior to passage-based ranking, we investigated five approaches to passage retrieval:

Okapi. According to [9], Okapi's native approach to passage retrieval works as follows. All passaging is done at search-time, not at index time. Passages are based on paragraph boundaries, and the experiments in this paper all use passages which are one paragraph in length. Given a query the retrieval engine first treats each document as a single passage and considers all documents whose weight exceeds a threshold set empirically at the weight of the 10,000th document. The documents above threshold are then broken into passages and each passage is scored. The initially retrieved documents are then re-ranked according to the score of their best passage, and the single best passage from each document is returned.

Approach 1. In this approach, all documents are pre-processed to produce a new document collection consisting of all passages drawn from the original document set which are then indexed. For consistency with Okapi, we again use paragraphs as passages. At search time the best passages are returned, possibly several from each document, in the order determined by the document ranking algorithm.

Approach 2. In this approach the top n retrieved full documents are post-processed into passages. For the $i\text{th}$ retrieved document $(i = 1, 2 \ldots n)$, a document collection is built from its passages, and a second stage retrieval is run against this collection, using the same retrieval engine as in the first stage, to determine the best passage from that document. The text of this passage is then returned as the $i\text{th}$ passage in the final ranking. Thus, full document retrieval is assumed to get the overall ranking right, but only the best passage from each document is selected for further processing.

Approach 3. In this approach, the top n retrieved full documents are again post-processed into passages, but this time, a single second-stage index is built from the passages from all n documents. The best n passages are then selected from this index, using the same retrieval engine as in the first stage, allowing multiple passages per document to be returned.

Approach 4. This approach is like approach 3, except that the second retrieval stage is limited to retrieve at most one passage from each original document. Thus, only one passage per document is returned, as in approach 2, but the ranking is determined by the passage score rather than the full document score. This approximates the Okapi approach, and is included primarily as a control, as non-Okapi-based tools were used to implement approaches 1–4 (see next section)[2].

[2] This approach is only an approximation, since, as noted above, the initial document retrieval stage in the Okapi model considers the top 10,000 documents, as opposed to 200 documents in approach 4.

Thus, to summarize, only approach 1 does index-time passaging, the other four approaches do search-time passaging. The differences between them are to do with whether the original ranking resulting from the initial query should guide the subsequent passage ranking (approach 2) or not (approaches 3 and 4 and Okapi) and whether one passage per document (Okapi, approaches 2 and 4) or multiple passages per document (approaches 1 and 3) should be returned.

Clearly these variations do not exhaust the space of possible approaches to passage retrieval. However, they provide an initial set to explore to see if significant differences in results begin to emerge.

4 Implementation

To run Okapi we simply downloaded the publicly available version[3] and used it as is.

To investigate the other approaches we used Lemur[4] as the underlying retrieval engine. Lemur has native support for the TREC document format, and supports vector-space, probabilistic and language modelling retrieval approaches against a single index. To keep experiments with Lemur as comparable as possible to those with Okapi we report here only the results of using the probabilistic approach (BM25 term weighting) within Lemur, as this is the model used in Okapi. We did investigate the other approaches supported by Lemur, but these had little significant effect.

To carry out passaging, a Perl program was written to read the source documents and split them into passages one paragraph in length. The line offsets of the passages within the original source files are stored in a flat index file to enable the passages to be reconstructed from the original data. The passages are output as TREC-formatted documents, which are then passed to Lemur for indexing. The index is built using the same list of stopwords as for Okapi.

There are some important practical issues of scalability that distinguish the pre-processing passaging approach (approach 1) from those that do passaging after initial retrieval. By treating every passage as an individual document, the pre-processing approach vastly increases both the space and time requirements of the indexing and retrieval programs. The subset of the TREC collection we used for testing (see next section) consists of 242,918 separate documents, and the full-document index built for the post-processing approaches required 1,122MB of disk space. The retrieval program took about 40 seconds to load the index, and at its peak, it consumed about 50MB of memory.[5]

In comparison, the preprocessor generated over 3.7 million separate passages (each a separate document to the IR system), a 15 fold increase. Though the index required only 1.4GB to store, the larger number of smaller documents meant that the retrieval program consumed 300MB of memory and took several

[3] http://www.soi.city.ac.uk/~andym/OKAPI-PACK

[4] http://www-2.cs.cmu.edu/~lemur/

[5] These results were obtained on a dual processor UltraSPARC, running Solaris 8, with 2GB of main memory.

minutes of intensive processing to load the index. In addition, the passage location index, used to map passage IDs back into the source text, required 130MB of space.

5 Experiments and Results

The test set used for these experiments was derived from the combined set of 1193 questions from TREC-9 (2000) and TREC 2001. These two evaluations operated over the same document set, and relevance judgments and answer patterns are available for both evaluations from NIST. The documents in the TREC collection are sourced from a variety of newswires by NIST, including the Associated Press (AP) newswire, the Wall Street Journal, the Los Angeles Times and the San José Mercury News. The documents are marked up in SGML, and the format of the markup varies from source to source. In particular, an algorithm to split documents into paragraphs for one source will not work for any of the other documents. In view of this, the experiments detailed below are based only on documents from one source. We chose the AP newswire, as 72% (863) of the 1193 test questions have at least one relevant document from this collection (i.e. a *correct* judgment with an AP document as the justification). The "next best" collection, in this sense, is the Los Angeles Times, for which only 53% of the questions have a relevant document. [6]

Each of the five approaches was evaluated by using each question in the question set as a query and returning the top 200 passages. For approaches which involved a two step process using Lemur (approaches 2, 3 and 4), 200 documents were retrieved in step one, then passaging was carried out and 200 passages were returned in step two in the manner of the specific approach. [7]

To inform our analysis of the results, we also calculated the actual redundancy, as defined in section 2, as follows. For each question we used the human assessors' judgments to pull out from the AP collection the documents identified as relevant to that question. For each of these documents we then split it into paragraphs, tested each of the NIST-supplied Perl patterns against each paragraph, and counted how many paragraphs matched at least one pattern. Actual redundancy for each question is then the total of these counts over all documents identified as answer bearing. Overall actual redundancy is the average of these redundancies per question. Note that this is still only an estimate of true redundancy because the assessors only confirm those documents as containing answers if they have been proposed by some system. Using this approach we determined that the actual redundancy is 14.3. This is the highest answer redundancy score

[6] There is some evidence that the AP newswire documents may not be representative of the whole collection – see [10].

[7] We restricted the maximum number of passages returned to 200 due to system limitations. Our QA system requires each retrieved passage to be stored as a separate file in the file system and the effect of moving beyond rank 200 (for 863 questions and five runs) was to run out of i-nodes on our Unix server. A more efficient representation is required to extend these experiments to lower ranks.

Table 1. Results of passage retrieval experiments – coverage

	% coverage at rank						
Run type	5	10	20	30	50	100	200
Okapi	48.78	60.02	66.63	69.76	74.51	78.79	82.04
Approach 1	43.80	54.58	63.50	67.67	71.38	78.22	83.43
Approach 2	45.89	55.39	63.73	67.21	72.31	76.25	79.72
Approach 3	41.48	54.11	63.85	68.48	73.70	80.07	85.52
Approach 4	40.79	52.26	60.37	65.47	69.06	74.39	77.87

Table 2. Results of passage retrieval experiments – answer redundancy

	Answer redundancy at rank						
Run type	5	10	20	30	50	100	200
Okapi	0.877	1.414	1.919	2.226	2.644	3.118	3.426
Approach 1	0.761	1.255	1.833	2.196	2.679	3.488	4.216
Approach 2	0.771	1.171	1.657	1.933	2.312	2.773	3.126
Approach 3	0.729	1.200	1.831	2.251	2.862	3.808	4.757
Approach 4	0.706	1.127	1.607	1.906	2.312	2.752	3.017

a system could achieve under our scoring system, if it retrieved every pattern-matching paragraph from every relevant document in the AP collection.

Tables 1 and 2, and the corresponding figure 1, show the results of the experiments. We see that the best coverage is consistently obtained by the native Okapi passage retrieval mechanism at the highest ranks, but that it is gradually overtaken by approaches 1 and 3 at lower ranks, though only marginally. However, Okapi has considerably lower answer redundancy scores than approaches 1 and 3 at these lower ranks. The most likely cause of this is that Okapi is limited to retrieving no more than one passage per document, and it seems highly likely, though we have not assembled the data to prove it, that in many cases the multiple answer instances contributing to the 14.3 actual redundancy figure fall within different passages in the same document. This conjecture is supported by the poor redundancy scores of approaches 2 and 4, which are also limited to returning only one passage per document. By contrast, those approaches which are able to retrieve multiple passages from the same document (approaches 1 and 3) demonstrate higher answer redundancy at all but the highest ranks. For the two approaches which can retrieve multiple passages per document, an examination of the average number of passages per source document retrieved per question reveals that approach 1 retrieves on average 1.2 passages per document, whereas approach 3 retrieves 1.6 – 33% more passages per document on average than approach 1.

Overall combined best performance in terms of coverage and answer redundancy, including passages down to rank 200, is obtained by approach 3. This is agreeable since approach 3 does not suffer from the space and time efficiency problems affecting approach 1. Of course while the approaches seem to be diverging at rank 200, we cannot rule out the possibility of their relative positions changing at even lower ranks.

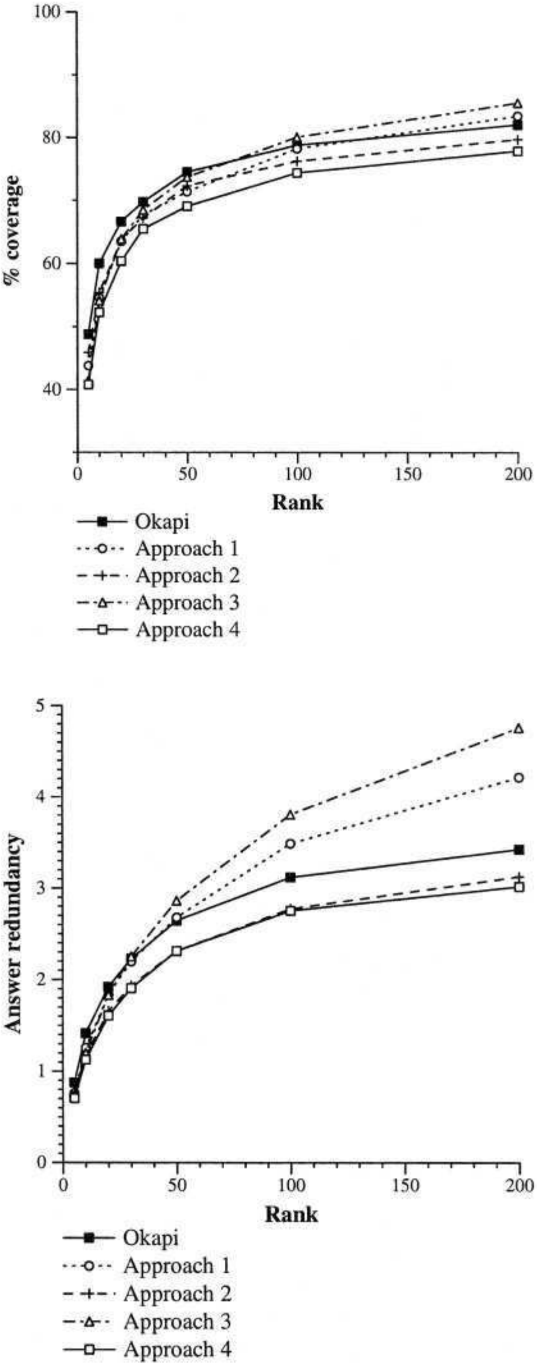


Fig. 1. Results of passage retrieval experiments

Finally, we should make two caveats. First, since the above observations are not informed by any statistical analysis of the differences between the approaches, they should be treated with caution. Second, the apparently positive effect on answer redundancy of being able to return multiple passages per document may be specific to the style or genre of the test collection. Further experimentation on broader classes of source material is required to see if this result generalises.

6 Related Work

To date, little work has been done on evaluating the effectiveness of passage retrieval approaches for question answering. LLopis, et. al [11] evaluated their passage retrieval system, using a measure similar to coverage as defined in this paper, though they did not formally define the measure. Tellex, et. al [12] carried out a detailed investigation of several different passage selection approaches to assess their effectiveness for QA. They also used a measure similar to coverage, though again it is not formally defined, and their systems were configured to return only 20 passages for each question – the effect of including either more or fewer passages was not explored. Neither work considered measures of answer redundancy.

The passage selection approaches considered by LLopis, et. al and Tellex, et. al are all variants of our approach 4, i.e. they first retrieve full documents, then retrieve the single best passage from each and order the results by similarity of the passage to the question. The work presented here is considerably wider in scope.

Monz [13] examined a variety of approaches to improving the performance of IR systems for QA, including the use of passage retrieval. However, the evaluation of these approaches is based simply on whether the full document from which the passage was drawn was judged relevant by a NIST assessor, and not on whether the passage itself contains an answer, and so the results are not directly comparable to those presented here.

7 Conclusions

We have investigated five approaches to paragraph-based passage retrieval for question answering, varying principally as to whether:

- they divide documents into passages prior to indexing, effectively treating each passage as an independent document, or after an initial retrieval stage;
- they permit only one or more than one passage per document to be returned;
- for search-time passaging approaches, the final passage ranking should be guided by the ranking of full documents resulting from the initial query or by the ranking obtained in the secondary passage retrieval stage.

To evaluate the utility of these approaches for question answering we have introduced two new measures, *coverage* and *answer redundancy* which capture what proportion of the question set has at least one answer returned in the top n passages and the average number of repetitions of the answer in the top n passages, respectively. These measures, we believe, are intuitive measures of the suitability of a passage retrieval approach for QA.

Applying these measures to assess five approaches to passage retrieval in one specific experiment using TREC QA data, we determined that the best-performing passage retrieval approach was one that first does full document retrieval, then splits the top retrieved documents into passages and performs a second passage retrieval operation against this passage set, returning the passages in the rank order determined by the second retrieval operation. This approach obtains both better coverage and answer redundancy scores beyond about rank 100.

A number of further questions immediately suggest themselves. Our experiment was restricted to the top 200 ranks. While the scores for the approaches appear to be diverging at this point, further experimentation at lower ranks should be carried out to confirm this. Of particular interest are the points at which coverage reaches 100% and answer redundancy approaches actual redundancy.

One would like to see higher coverage and redundancy at higher ranks. Can this be achieved using other passage retrieval approaches not explored here? Or, are current performance levels unsurpassable, given an approach which uses the raw question words as the query to the retrieval system? Various approaches to query "enhancement" need to be considered.

While higher coverage and answer redundancy would appear to be inherently good for QA systems, there may a critical tradeoff between specific values for coverage and redundancy and the rank at which are these are obtained. For example, a QA system may do better with the top 50 passages than with the top 100, even though the top 100 have higher coverage and redundancy, simply because of the "noise" introduced by a further 50 passages. The interaction between coverage and redundancy at certain ranks and the answer extraction capabilities of QA systems needs to be investigated.

Acknowledgements. The authors would like to thank the anonymous reviewers of this paper for their comments.

References

1. Voorhees, E.M.: Overview of the TREC-2001 question answering track. In: NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001). (2001)
2. Milward, D., Thomas, J.: From information retrieval to information extraction. In: Proceedings of the ACL Workshop on Recent Advances in Natural Language Processing and Information Retrieval. (2000) Available at: http://www.cam.sri.com/html/highlight.html.

3. Molla Aliod, D., Berri, J., Hess, M.: A real world implementation of answer extraction. In: Proceedings of the 9th International Conference on Database and Expert Systems Applications Workshop "Natural Language and Information Systems" (NLIS'98). (1998) 143–148
4. Light, M., Mann, G.S., Riloff, E., Breck, E.: Analyses for elucidating current question answering technology. Natural Language Engineering **7** (2001) 325–342
5. Brill, E., Lin, J., Banko, M., Dumais, S., Ng, A.: Data-intensive question answering. In: NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001). (2001) 393–400
6. Callan, J.P.: Passage-level evidence in document retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and Development in Information Retrieval. (1994) 302–310
7. Voorhees, E.M., Tice, D.M.: Building a question answering test collection. In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and Development in Information Retrieval. (2000) 200–207
8. Scott, S., Gaizauskas, R.: University of Sheffield TREC-9 Q & A System. In: Proceedings of The Ninth Text REtrieval Conference (TREC 9), NIST Special Publication 500-249 (2000) 635–644
9. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at TREC-3. In: NIST Special Publication 500-225: The Third Text REtrieval Conference (TREC-3). (1994) 109–126
10. Cavnar, W.B.: N-gram based text filtering for TREC-2. In: NIST Special Publication 500-215: The Second Text REtrieval Conference (TREC-2). (1993) 171–179
11. LLopis, F., Vicedo, J.L., Ferrández, A.: Passage selection to improve question answering. In: Proceedings of the COLING 2002 Workshop on Multilingual Summarization and Question Answering. (2002)
12. Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G.: Quantitative evaluation of passage retrieval approaches for question answering. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and Development in Information Retrieval. (2003) 41–47
13. Monz, C.: Document retrieval in the context of question answering. In: Proceedings of the 25th European Conference on Information Retrieval Research (ECIR-03). (2003) 571–579

Identification of Relevant and Novel Sentences Using Reference Corpus

Hsin-Hsi Chen, Ming-Feng Tsai, and Ming-Hung Hsu

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
hh_chen@csie.ntu.edu.tw, {mftsai,mfhsu}@nlg.csie.ntu.edu.tw

Abstract. The major challenging issue to determine the relevance and the novelty of sentences is the amount of information used in similarity computation among sentences. An information retrieval (IR) with reference corpus approach is proposed. A sentence is considered as a query to a reference corpus, and similarity is measured in terms of the weighting vectors of document lists ranked by IR systems. Two sentences are regarded as similar if they are related to the similar document lists returned by IR systems. A dynamic threshold setting method is presented. Besides IR with reference corpus, we also use IR systems to retrieve sentences from given sentences. The corpus-based approach with dynamic thresholds outperforms direct retrieval approach. The average F-measure of relevance and novelty detection using Okapi system was 0.212 and 0.207, 57.14% and 58.64% of human performance, respectively.

1 Introduction

How to obtain relevant information from a considerable amount of data collection has become increasingly important. Current information retrieval (IR) systems only return documents satisfying users' information needs, but they do not precisely locate the relevant sentences. Therefore, users have to go through the whole documents to find the relevant information. Moreover, traditional IR systems do not identify which sentences contain new information. Filtering redundant information out and locating novel information is indispensable for some emerging applications like summarization and question-answering [4].

There are some sorts of relevance and novelty detection on document level in Topic Detection and Tracking (TDT) [2]. Link detection relates news stories on the same topic [3] and first story detection tries to identify the first article with a new event. Novelty track in TREC 2002 [5] is the first attempt to locate relevant and new sentences instead of the whole documents containing duplicate and extraneous information. Similarity computation is a fundamental operation for relevance and novelty judgment on both sentence and document levels. However, the amount of information of a sentence that can be used in similarity computation is much fewer than that of a document. That forms the major challenging issue.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 85–98, 2004.

In the past, word matching and thesaurus expansion were adopted to recognize if two sentences touched on the same theme in multi-document summarization [4]. Such an approach has been employed to detect the relevance between a topic description and a sentence [8]. The similarity computation can also be performed by an information retrieval system. Zhang *et al.* [11] employed an Okapi system to retrieve relevant sentences with a topic description, and a fixed heuristic threshold was adopted. Larkey *et al.* [6] studied how many sentences were relevant in different size of documents. Allan *et al.* [1] focused on the novelty detection algorithms and showed how the performance of relevant detection affects that of novelty detection. Instead of using an IR system to select relevant sentences directly, an external corpus can be consulted [8]. Both a topic description and a sentence are considered as queries to the reference corpus through an IR system. Two sentences are relevant if similar sets of relevant documents are retrieved.

This paper shows how to extract relevant sentences from several known relevant documents, and how to determine new sentences from the extracted relevant sentences. The decision about what information is new depends on the order of the occurrence of the information. In other words, "a novel sentence" means that all of the relevant information in this sentence is never covered by the relevant sentences delivered previously. Section 2 presents a concept matching approach, a test set and evaluation metrics. Section 3 uses reference corpus and IR systems. The effects of different issues, including with/without reference corpus, static/dynamic settings of thresholds, and various IR systems, are compared. Section 4 extracts novel sentences from relevant sentences. Section 5 concludes the remarks.

2 A Concept Matching Approach

The problem of novelty task is defined as follows:

> *Given a topic description and a sequence of sentences, a novelty detection system should identify which sentences are relevant to the topic description, and which sentences are novel relative to the other sentences under a specific topic.*

The original sequence of sentences is called *given sentences,* and the resulting two lists are called *relevant sentences* and *novel sentences.* The given sentences came from some relevant documents. A novelty task is composed of two major components, i.e., a relevance detector and a novelty detector. The relevance detector receives a sequence of sentences from known relevant documents, and determines which sentences are on topic. Those relevant sentences will be delivered to the novelty detector and the redundant sentences will be filtered out. The remaining sentences are *novel* and *relevant.* Relevant detector is very important because its performance will affect the performance of a novelty detector.

A relevance detector attempts to identify those sentences containing the relevant information from the known relevant documents. The key issue behind relevance detection is how to measure the similarity of a topic description and the given sentences. Because the basic unit of similarity measure is a sentence instead of the whole document, we have to deal with the problem of the lack of information within a sentence during distinguishing relevant and irrelevant sentences. A concept matching approach is proposed for relevance detection. A predicate and its surrounding arguments form a kernel skeleton in a sentence, so that verbs and nouns are important features for similarity computation. In this way, all the given sentences are tagged by using a part-of-speech tagger. After tagging, nouns and verbs are extracted. Then WordNet is applied to find the synonymous terms for concept matching. Noun and verb taxonomies with hyponymy/hypernymy relations are consulted. The similarity of two sentences is in terms of noun-similarity and verb-similarity as follows.

$$noun\_sim(s1, s2) = \frac{m}{\sqrt{ab}} \tag{1}$$

$$verb\_sim(s_1, s_2) = \frac{n}{\sqrt{cd}} \tag{2}$$

$$sim(s_1, s_2) = noun\_sim(s_1, s_2) + verb\_sim(s_1, s_2) \tag{3}$$

where s_1 and s_2 denote two sentences, respectively; m and n denote the number of concept matching for nouns and verbs, respectively; a and b are the total number of nouns in s_1 and s_2, respectively; and c and d are the total number of verbs in s_1 and s_2, respectively.

Total 49 topics and 49 sets of given sentences in TREC 2002 Novelty track [5] are applied to evaluate the performance of relevance detector. Precision, recall and F-measure shown as follows are employed.

Recall (R) = #RELEVANT matched / #RELEVANT (4)

Precision (P) = #RELEVANT matched / #sentences submitted (5)

F-measure (F) = 2 Recall * Precision / (Recall + Precision) (6)

Average F-measure = $\sum$ F-measure / #TOPIC (7)

When the threshold is set to 0.4, the average F-measure of the concept matching approach is 0.125. Besides, a baseline model that randomly selects sentences from the given sentences is also adopted for comparison. The average F-measure of the baseline model was 0.040, and the average F-measure of human judge was 0.371 [5]. The experiments show that the performance of the concept matcher is better than that of the baseline model, but is still far less than that of human being. The outside resource, i.e., WordNet, seems not to be enough to measure the similarity in these experiments. In the following we will consult another resource – say, a reference corpus.

3 Relevance Detection Using IR Approach

3.1 IR with Reference Corpus

To use a similarity function to measure if a sentence is on topic is similar to the function of an IR system. We use a reference corpus, and regard a topic and a sentence as queries to the reference corpus. An IR system retrieves documents from the reference corpus for these two queries. Each retrieved document is assigned a relevant weight by the IR system. In this way, a topic and a sentence can be in terms of two weighting vectors. Cosine function measures their similarity, and the sentence with similarity score larger than a threshold is selected. The issues behind the IR with reference corpus approach include the reference corpus, the performance of an IR system, the number of documents consulted, the similarity threshold, and the number of relevant sentences extracted.

The reference corpus should be large enough to cover different themes for references. In the experiments, the document sets used in TREC-6 text collection [10] were considered as a reference corpus. It consists of 556,077 documents. Two IR systems, i.e., Smart [9] and Okapi [7], were adopted to measure the effects of the performance of an IR system. In the initial experiments, Smart system with the basic setting (i.e., *tf*idf* scheme without relevance feedback) was employed. It had average precision 0.1459 on the TREC topics 301-350. Okapi was in the option of bm25, and had average precision 0.2181 on the same document set.

3.2 How Many Documents Reported

How many documents should be reported by an IR system is an important issue for similarity measurement between a topic and a given sentence. Both relevant and irrelevant documents may be reported in the result list. That depends on the IR performance. The effects of the sizes of resulting document lists were investigated. Table 1 summarizes the results of using Smart and Okapi when the threshold is set to 0.1. The first column shows that different number of documents, i.e., 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 documents, are returned by Smart and Okapi, respectively.

It shows that smaller result list (e.g., 50 documents) is better than larger result list when Smart system is adopted. This is because the relevant document set is comparatively much smaller than the irrelevant document set for a query, and the irrelevant documents in the two result lists tend to be different. Smaller result list decreases the possibility to incorporate different irrelevant documents, but also decreases the possibility to find out the same relevant documents. Enlarging the result list means the number of the same relevant documents may be increased, but different irrelevant documents are also added. In contrast, the performance of Okapi-based system is increased from reporting 50 documents till 250 documents. After that, the performance starts to decrease. This is because Okapi outperforms Smart. Larger result list (within 250 documents) covers more relevant documents. In the experiments, the best average F-measures, 0.170 and 0.176, were achieved when the sizes of result list were 50 and 250 documents by using Smart and Okapi, respectively.

3.3 Threshold Setting

We also made experiments with different thresholds (between 0 and 0.3), and smaller number of returned documents. Figures 1 and 2 show the experimental results. The best F-measures of using Smart and Okapi are 0.175 and 0.182, respectively. Because we did not employ the distribution of similarity scores, the thresholds were "guessed", and the thresholds were fixed in different topics. That is unfair in some cases.

Table 1. Effects of Size of Returned Documents

Number of consulted documents	Smart-based			Okapi-based		
	Avg. P	Avg. R	Avg. F	Avg. P	Avg. R	Avg. F
50	0.13	0.4	**0.170**	0.15	0.49	0.169
100	0.13	0.43	0.154	0.15	0.48	0.174
150	0.12	0.46	0.144	0.13	0.49	0.174
200	0.11	0.48	0.137	0.14	0.48	**0.176**
250	0.11	0.50	0.137	0.13	0.49	**0.176**
300	0.10	0.51	0.135	0.13	0.49	0.173
350	0.10	0.52	0.130	0.12	0.49	0.170
400	0.10	0.54	0.127	0.12	0.50	0.171
450	0.09	0.54	0.124	0.12	0.50	0.170
500	0.09	0.55	0.120	0.11	0.51	0.169

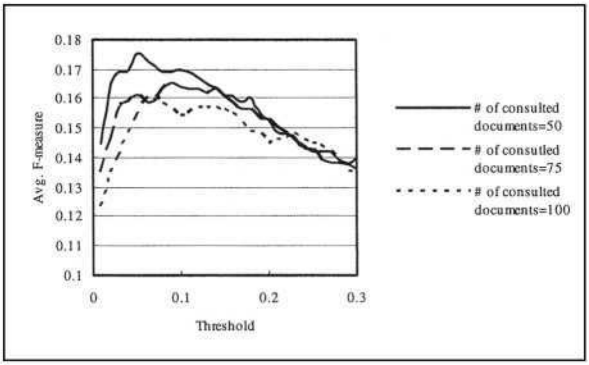


Fig. 1. Effects of Fixed Thresholds Using Smart

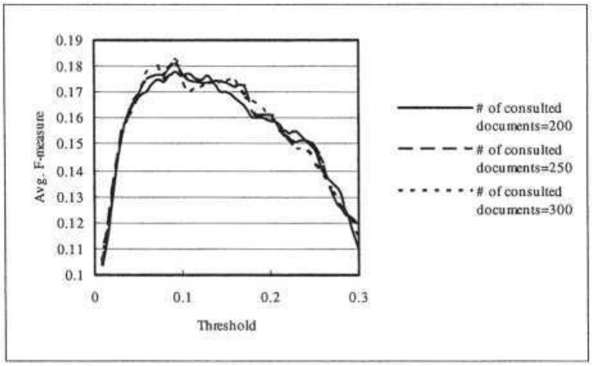


Fig. 2. Effects of Fixed Thresholds Using Okapi

A threshold setting model is proposed as follows to deal with this problem. Assume normal distribution with mean μ and standard deviation σ is adopted to specify the similarity distribution of the given sentences with a topic. We compute the cosine of a topic vector T and a given sentence vector S_i ($1 \leq i \leq m$) as below, where m denotes total number of the given sentences. The percentage n denotes that top n percentages of the given sentences will be reported. Similarity thresholds ($\mathrm{TH}_{\mathrm{relevance}}$) are determined by these percentages.

$$\mu = \frac{\sum_{i=1}^{m} \cos(T, S_i)}{m} \tag{8}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{m} (\cos(T, S_i) - \mu)^2}{m}} \tag{9}$$

$$\mathrm{TH}_{\mathrm{relevance}} = \mu + z\sigma \tag{10}$$

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-y^2/2} dy = 1 - n \tag{11}$$

Figure 3 shows that total n (%) of given sentences fall in the gray area are considered as relevant. z is equal to 1.282, 0.84, 0.524, 0.253 and 0 when n is 10%, 20%, 30%, 40%, and 50%, respectively.

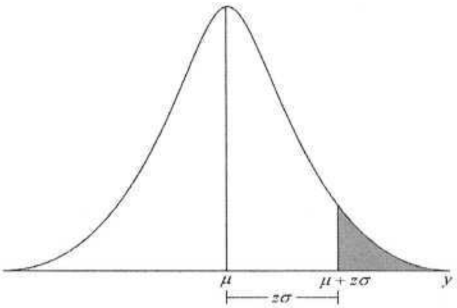


Fig. 3. Normal Distribution with Mean μ and Standard Deviation σ

Various settings of n (percentage) were experimented, and the results using Smart and Okapi are listed in Figures 4 and 5, respectively. Smart-based relevance detector achieves better performance when larger percentage of sentences is selected. On the contrary, the larger the percentage is, the worse the performance is, when some critical point is reached using Okapi. The major reason is: Okapi gets better retrieval performance than Smart, so that it pulls the relevant sentences in the front of normal distribution. The best F-measures are 0.190 and 0.206. Using n (%) to determine the thresholds is a dynamic approach, which is better than static threshold approach.

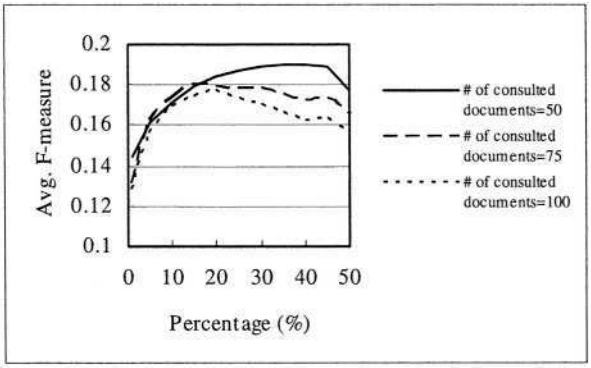


Fig. 4. Effects of Fixed Percentages Using Smart

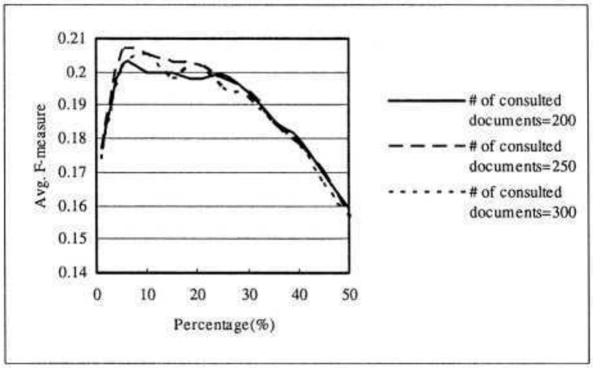


Fig. 5. Effects of Fixed Percentages Using Okapi

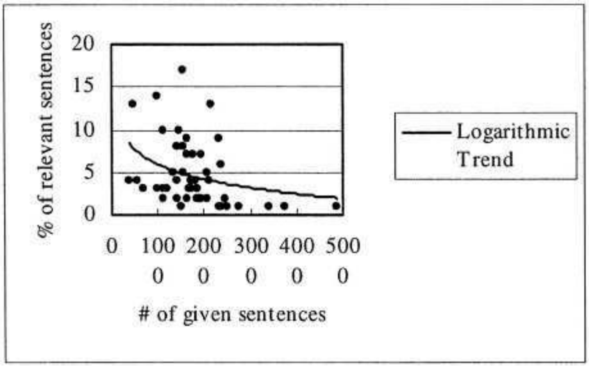


Fig. 6. An illustration of Logarithmic Trend

Table 2. Effects of Dynamic Percentage

Number of consulted documents	Smart-based			Okapi-based		
	50	75	100	200	250	300
Ln-1	0.164	0.167	0.163	0.203	0.208	**0.212**
Ln-2	0.177	0.180	0.176	0.204	0.207	0.205
Ln-3	0.185	0.178	0.176	0.204	0.205	0.205
Ln-4	0.189	0.179	0.174	0.200	0.201	0.198
Ln-5	**0.191**	0.181	0.172	0.194	0.191	0.191

Even though the above dynamic approach has better performance, it is still "fixed percentage" for all topics. We consider further how to select "good" percentages for individual topics. Larkey *et al.* [6] showed that only 5% of the sentences contained relevant materials for average topic. From their collection statistics [6], we used logarithmic regression as follows to simulate the relationship between total number of the given sentences and number of the relevant sentences. Figure 6 illustrates the trend.

$$n = -2.4938Ln(x) + 23.157 \tag{12}$$

where x is total number of given sentences, and n is the suggested percentage.

After computing n using Formula (12), we derived z using Formula (11) and finally $\mathbf{TH}_{\mathbf{relevance}}$ using Formula (10).

Table 2 summarizes the F-measures of using dynamic percentage by Smart and Okapi, respectively. Dynamic percentage is better than fixed percentage. The best performance of dynamic percentage using Smart is 0.191 when the size of consulted documents is set to 50 and logarithmic metric is multiplied by 5, which gets about 1 % improvement to the fixed percentage. The best F-measure of dynamic percentage using Okapi is 0.212, when the size of consulted documents is set to 300 and the original logarithmic metric is employed. It gets about 3% increases to the fixed percentage experiments.

The best performance among these experiments is 0.212, i.e., 57.14% of human performance (0.371). Figure 7 lists the performance of each topic when the number of consulted documents is 300 using Okapi system. Two dotted lines, i.e., one is human performance (0.371) and the other one is baseline performance (0.040), are provided for reference. Performance of our system in 6 topics (358, 364, 365, 368, 397, and 449) is competitive to that of human judge. In contrast, performance in 3 topics (377, 420, and 432) is lower than that of random selection. The average F-measure of the remaining 40 topics are below human performance, but better than that of baseline model.

3.4 IR without Reference Corpus

Even using an IR system, we have two alternatives to select the relevant sentences, i.e., with and without a reference corpus. In the corpus-free approach, the given sentences form a database itself, and a topic is submitted to an IR system to retrieve the similar sentences directly. The resulting sentences ranked and reported by the IR system are called *candidate sentences.* A dynamic percentage of candidates with higher scores will be reported as relevant. The percentage and the relevant thresholds are determined in the similar way as the corpus-based approach. The best F-measures of IR approach without reference corpus are 0.113 and 0.165, respectively. Smart-based and Okapi-based systems without reference corpus decrease 10% and 9% performance, respectively.

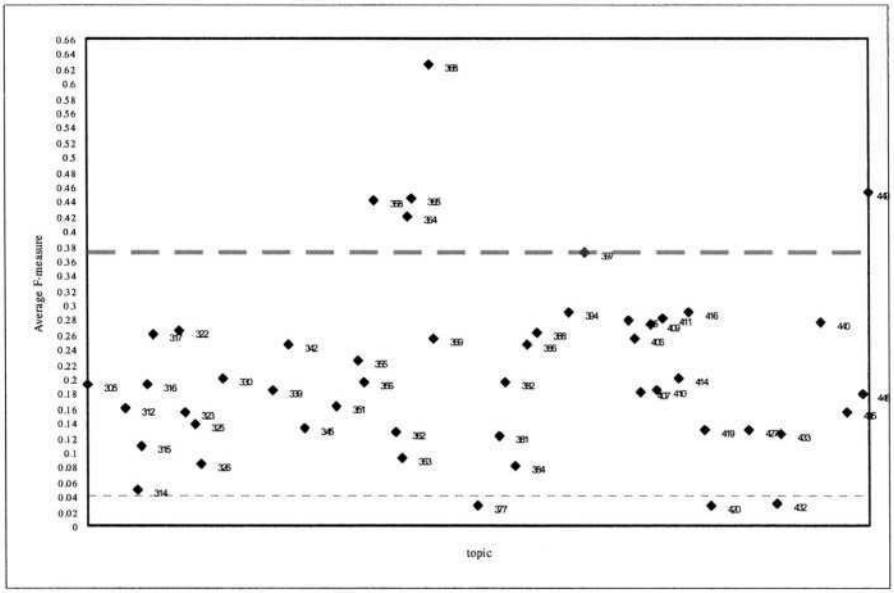


Fig. 7. Average F-measure of Relevance Detection for Each Topic

4 Novelty Detection Using Reference Corpus

Novelty detector identifies new information among the sentences extracted by the relevance detector. In other words, novelty detector will filter out the redundant sentences among the relevant sentences. The key issue on the detection of new information is how to differentiate the meaning of sentences accurately. Sentences may contain too less information to distinguish their differences, so that certain information expansion method is required.

We extend the idea in Section 3, i.e., employing a reference corpus to select the relevant information, to find the relationship among relevant sentences. Similarly, we use the same reference corpus and regard each relevant sentence as a query to this corpus. Documents in the corpus are ranked by an IR system, and the documents with higher scores are reported for each relevant sentence. Each retrieved document is assigned a weight, in such a way that a sentence is still represented as a vector. Cosine function measures the similarity of any two sentences. Two sentences are regarded as similar if they are related to the similar document lists.

On the one hand, the cosine value of two sentences indicates that how similar they are. On the other hand, the higher value indicates one sentence is somewhat redundant relative to the other sentence. A threshold of novelty decision, $\mathbf{TH}_{\mathbf{novelty}}$, determines the degree of redundancy. If the similarity score of two sentences is larger than $\mathbf{TH}_{\mathbf{novelty}}$, then one of them has to be filtered out depending to their temporal order. In this way, the redundant sentences are filtered out and only the novel sentences are kept. The remaining sentences are the result of the novelty detector.

Two algorithms are proposed as follows to deal with the novelty detection problem. Assume there are *r* relevant sentences, s_1, s_2, ..., s_r for topic *t*.

(1) Static threshold approach
Let *T* be a set containing novel sentences found up to know. Initially, $T=\{s_1\}$. For each relevant sentence s_i $(2 \le i \le r)$, if there exists a sentence in *T* whose similarity with s_i is larger than a predefined threshold, then s_i is not a novel sentence and is removed; otherwise, s_i is kept in *T*.

(2) Dynamic threshold approach
Assume s_1 is a novel sentence. Compute the similarities between s_1 and s_i $(2 \le i \le r)$. Determine the novelty threshold, $\mathbf{TH}_{\mathbf{novelty}}$, in the same way as $\mathbf{TH}_{\mathbf{relevance}}$. Filter out the top *n*% of sentences with the higher similarities with s_1. Let R be the remaining sentences. If the number of sentences in *R* is less than 30[1] then regard these sentences as novel sentences and stop. Otherwise, select the first sentence in R, regard it as a novel sentence and repeat the same filtering task.

We chose the results from the best relevance detectors mentioned in last section, i.e., Smart-based and Okapi-based systems with average F-measure 0.191 and 0.212, to test these two approaches. The performance of static threshold approach is shown in Figure 8. Okapi-based novelty detector still outperforms Smart-based novelty detector. Besides, it also indicates that more sentences are filtered out when $\mathbf{TH}_{\mathbf{novlety}}$ is lower. The performance increased as $\mathbf{TH}_{\mathbf{novelty}}$ increased. Using higher novelty threshold, two sentences should have much higher similarity to pass the threshold if they are similar. The lower the probability two sentences pass the threshold, the higher the probability both sentences are novel. Figure 9 illustrates the results of dynamic threshold approach. When more percentages of sentences are filtered out, the performance of both Smart-based and Okapi-based novelty detectors are decreased.

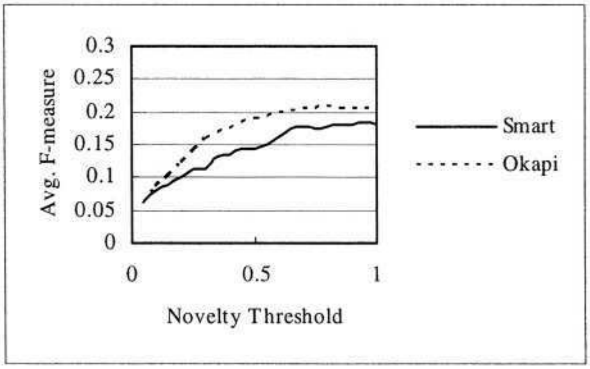


Fig. 8. Results of Static Novelty Threshold

[1] A sample size of at least 30 has been found to be adequate for normal distribution.

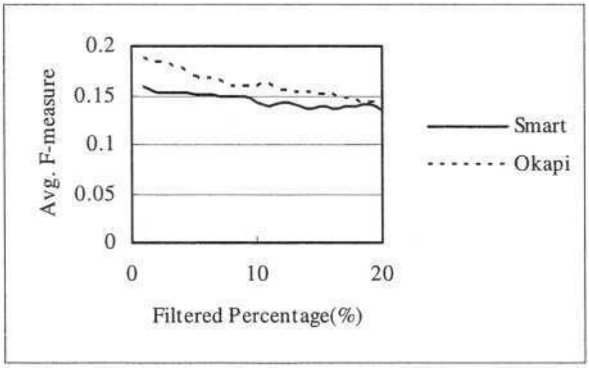


Fig. 9. Results of Dynamic Novelty Threshold

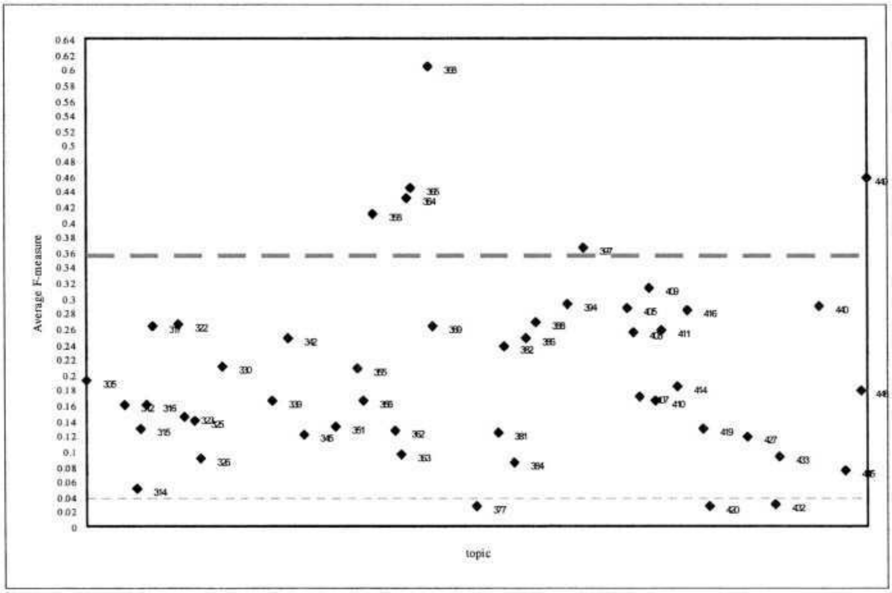


Fig. 10. Further Examination of the Best Novelty Detection

The best performance among these experiments is 0.207, when the novelty threshold is set to 0.8 statically, and total 300 documents reported by Okapi are consulted. Figure 10 examines the performance of each topic furthermore. Two dotted lines, one for human performance (0.353) and the other one for baseline performance (0.036), are provided for reference. Performance of our approach in 6 topics (i.e., 358, 364, 365, 368, 397, and 449) is competitive to that of human judge. In contrast, performance in 3 topics (377, 420, and 432) is lower than that of the baseline model. The average F-measure of the remaining 40 topics are below human performance, but better than that of baseline model.

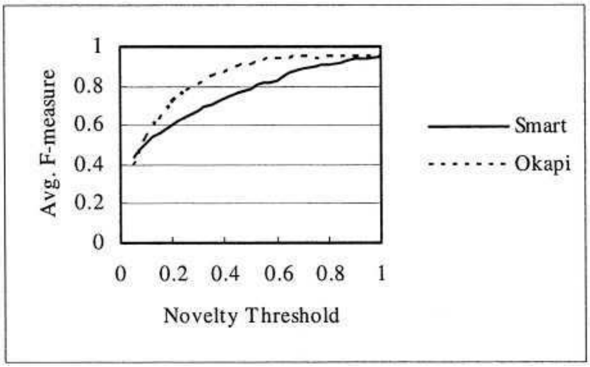


Fig. 11. Ideal Performance of Static Novelty Threshold Approach

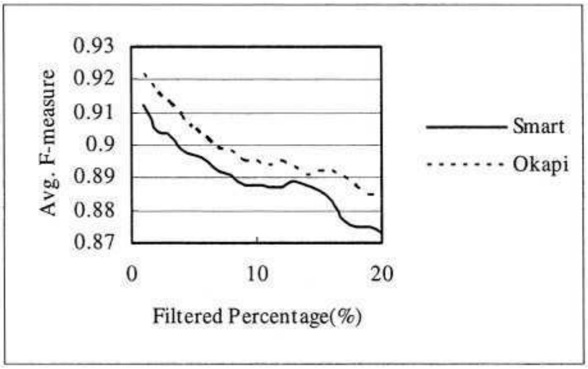


Figure 12. Ideal Performance of Dynamic Novelty Threshold Approach

In general, the average F-measure of the novelty detector is better than that of the baseline model (i.e., 0.036). However, the performance is still not comparable to the human assessors (i.e., 0.353). It only achieves 58.64% of human performance. The major reason is that the result of relevance detector contains irrelevant sentences, so that novelty detector false identifies that those irrelevant sentences contain new information. As mentioned before, the relevance part is more difficult to be overcome in this task.

We also conducted another set of experiments to evaluate the ideal performance of locating novel sentences. These experiments take correct relevant information as input to novelty detector, so that there are no propagation errors from relevant detectors. Figure 11 shows the results of static novelty threshold approach. The performance was increased when $\mathbf{TH}_{novelty}$ was increased. This is because more sentences are filtered out when $\mathbf{TH}_{novelty}$ is lower. The ideal performance of Okapi-based novelty detector is 0.945 and the performance is above 0.912 when threshold is larger than 0.5. Figure 12 shows the results of dynamic novelty threshold approach. The ideal performance of the Okapi-based system is 0.922. The average F-measure dropped quickly, when more percentage of sentences are filtered out.

5 Conclusions and Future Work

This paper proposed concept matching and IR approaches to identify sentences that are novel and redundant as well as relevant and irrelevant. Although the method of matching noun and verb keywords and the related expansion achieved average F-measure 0.125, which is better than the baseline performance (i.e., 0.040), words in sentences are still not enough for the relevance detection. We presented an information expansion using a reference corpus to deal with this problem. We postulated that if two sentences have the similar meaning, then their behavior on information retrieval to the reference corpus is similar. Logarithmic regression approximates how many percentages of sentences are relevant for each topic. This value determines an offset from mean in normal distribution and thus the similarity threshold. That forms a rigid procedure instead of heuristics to determine the needed parameters. The experiment results show that Okapi-based relevant detector with dynamic threshold setting, which depend on topics and given sentences, are better than the other approaches. The best average F-measure of relevance detector is 0.212, which is 57.14% of human performance (0.373). When the idea was extended to novelty detector, the average F-measure is 0.207, which is 58.64% of human performance (0.353). The effects of the IR systems, e.g., query construction and relevance feedback, will be investigated. Besides, the deep syntactic and semantic analysis of sentences to distinguish relevant and novel sentences will be explored.

References

1. Allan, J., Wade, C., and Bolivar, A.: Retrieval and Novelty Detection at the Sentence Level. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Toronto, Canada, July 28–August 01, 2003. ACM (2003) 314-321
2. Allan, J., Carbonnell, J., and Yamron, J.: Topic Detection and Tracking: Event-Based Information Organization. Kluwer (2002)
3. Chen, H.H., and Ku, L.W.: An NLP & IR Approach to Topic Detection. In Topic Detection and Tracking: Event-Based Information Organization, James Allan, Jaime Carbonnell, Jonathan Yamron (Editors). Kluwer (2002) 243-264
4. Chen, H.H., Kuo, J.J., Huang, S.J., Lin, C.J., and Wung, H.-C.: A Summarization System for Chinese News from Multiple Sources. In Journal of American Society for Information Science and Technology. (2003)
5. Harman, D.: Overview of the TREC 2002 Novelty Tree. In Proceedings of the Eleventh Text REtrieval Conference. NIST Special Publication: SP 500-251, Gaithersburg, Maryland, November 19-22, 2002. TREC (2002)
6. Larkey, L. S. et al.: UMass at TREC2002: Cross Language and Novelty Tracks. In Proceedings of the Eleventh Text REtrieval Conference. Gaithersburg, NIST Special Publication: SP 500-251, Gaithersburg, Maryland, November 19-22, 2002. TREC (2002)
7. Robertson, S.E., Walker, S., and Beaulieu, M.: Okapi at TREC-7: Automatic ad hoc, Filtering, VLC and Interactive. In Proceedings of the Seventh Text REtrieval Conference, Gaithersburg, NIST Special Publication: SP 500-242, Gaithersburg, Maryland, November 9-11, 1998. TREC 7 253-264.
8. Tsai, M.F., and Chen, H.H.: Some Similarity Computation Methods in Novelty Detection. In Proceedings of the Eleventh Text REtrieval Conference. Gaithersburg, NIST Special Publication: SP 500-251, Gaithersburg, Maryland, November 19-22, 2002. TREC (2002)

9. Salton, G., and Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. In Information Processing and Management. Vol. 5, No. 24, pp. 513-523.
10. Voorhees, E.M., Harman, D.K. (Eds.) Proceedings of the Sixth Text Retrieval Conference. NIST Special Publication: SP 500-240, Gaithersburg, Maryland, November 19-21, (1997)
11. Zhang, M. et al.: THU at TREC2002: Novelty, Web and Filtering. In Proceedings of the Eleventh Text REtrieval Conference, NIST Special Publication: SP 500-251, Gaithersburg, Maryland, November 19-22, 2002. TREC (2002).

Answer Selection in a Multi-stream Open Domain Question Answering System

Valentin Jijkoun and Maarten de Rijke

Language & Inference Technology Group, University of Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam, The Netherlands
{jijkoun,mdr}@science.uva.nl

Abstract. Question answering systems aim to meet users' information needs by returning exact answers in response to a question. Traditional open domain question answering systems are built around a single pipeline architecture. In an attempt to exploit multiple resources as well as multiple answering strategies, systems based on a multi-stream architecture have recently been introduced. Such systems face the challenging problem of having to select a single answer from pools of answers obtained using essentially different techniques. We report on experiments aimed at understanding and evaluating the effect of different options for answer selection in a multi-stream question answering system. We examine the impact of local tiling techniques, assignments of weights to streams based on past performance and/or question type, as well redundancy-based ideas. Our main finding is that redundancy-based ideas in combination with naively learned stream weights conditioned on question type work best, and improve significantly over a number of baselines.

1 Introduction

Question answering is a variation on the traditional document retrieval task where, in response to a user's question, an answer has to be returned instead of a ranked list of relevant documents from which the user has to extract an answer herself. Traditional question answering systems typically consist of a single processing stream that performs three steps in a sequential fashion: question analysis, search, and answer selection [10,14,16,17]. These systems typically focus on the corpus from which the answers are to be extracted, and they may use a small number of additional resources, either unstructured (such as the Web), semi-structured (such as WordNet or the CIA World Fact Book), or structured (such as geography databases). Essentially, such single-stream systems adopt a "one size fits all" approach, treating factoid questions of all types in the same manner.

Recently, a number of teams have adopted more complex architectures for their question answering systems, either involving feedback loops as part of a single stream architecture [15], or involving multiple streams that somehow implement different answering strategies [1,2,4,11]. The motivations underlying

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 99–111, 2004.

these multi-stream approaches are two-fold: 1. Some answering strategies may be highly effective for certain question types, but not for others. 2. An important practical benefit of multi-stream architectures is easy modification, maintenance, and testing of the different subsystems as well as easy integration of multiple source of information.

One of the big challenges raised by multi-stream approaches to open domain question answering is that at some stage a global choice needs to be made: which of the many candidate answers produced by the independent streams is to be chosen as *the answer* to be returned by the system? The important thing to notice here is that techniques and information sources used by different streams can be very different: some can be more reliable than others and the scoring methods of the streams may be incompatible. Think, for instance, of a Web answering stream based on redundancy and a knowledge base lookup stream. In such a combined system, the task of reducing answer candidates "to a common denominator" is highly non-trivial.

In this paper we experiment with a number of answer selection strategies in a multi-stream question answering environment. We generalize and develop the methods described earlier [1,4], present novel techniques for combining answer streams and systematically evaluate the performance of different methods in different combinations, using our own multi-stream question answering system for a case study. One of our main findings is that, using a fairly limited amount of data, a relatively simple machine learning method performs just as well as humans in assigning weights to streams, measured in terms of the performance of the overall system.

The selection methods we consider do not make any assumptions about the precise nature of the streams involved, or even about the answer types being used in question classification. All we need is data on the past performance of the streams. For these reasons, we believe that the methods and results that we present below are widely applicable in the question answering domain.

In the remainder of this paper, we first discuss related work on answer selection and result merging. We then describe the architecture of one particular multi-stream question answering architecture. Next, we discuss in detail different aspects of answer selection in such architectures. Finally, we report on experiments with different answer selection schemes and discuss our findings.

2 Related Work

In a single-stream question answering environment, the goal of answer selection is to choose from a pool of answer candidates the most likely answer for the question. There are many approaches to answer selection in the single-stream setting. Here, we only have space to mention a few. In their TREC 2002 system, the BBN team used the standard single-stream pipeline by using a document retrieval system to select documents that are likely to contain answers to a given question and then ranking candidate answers based on the answer contexts using the same retrieval system [20]. They then used a few constraints to re-rank the candidates; these constraints include whether a numerical answer quantifies the correct noun, whether the answer is of the correct location sub-type and whether the answer satisfies the verb arguments of the question. Like BBN's question answering system, the system developed by LCC is an example of a single-stream architecture whose answer selection process is very knowledge-intensive [15]. It incorporates lots of AI-like technology, by actually attempting to prove candidate answers from text, with a number of feedback loops and sanity-checking that can reject answers or require additional checking. By way of contrast to these knowledge-intensive selection and reranking methods in single-stream architectures we mention a purely data-driven approach due to Magnini et al. [13]. They employ the redundancy of the Web to re-rank answer candidates found in the collection, by using hit counts for question and answer terms.

Let's turn to multi-stream question answering environments now. Here, the task of selecting the final answer is complicated by the fact that the final answer has to be selected from several pools of ranked candidates found by different streams. We need to compare answers coming from different, often incomparable sources and pick the one that is most likely to be correct.

IBM's PIQUANT question answering system used at TREC 2002 [1] implements a multi-strategy and multi-source approach. It faces a problem of combining answers coming from external resources (such as the Web) and answers extracted from the reference corpus. The question answering evaluation methodology of TREC requires that all answers be justified in the corpus, so the proposed solution is to use answer feedback: for each external answer candidate the best (if any) relevant passage in the reference corpus is found and used for candidate ranking. Because of this answer feedback, all answering streams produce ranked candidate answers in a uniform fashion, which obviously simplifies the final answer selection.

Some multi-stream systems solve the answer selection problem based solely on the stream that produced it. E.g., the University of Waterloo's MultiText system consults multiple resources, using a variety of methods, ranging from shallow parsing to word n-gram mining [2]. In MultiText, these approaches are organized in two streams, one called the early-answering subsystem (which consults ready-made databases), the other called statistical answer-selection (which resembles the traditional single-stream question answering pipeline). Final answer selection is based purely on the stream that delivered the answer: whenever the early-answering stream produces an answer, MultiText judges it to be correct, based on the fact that whenever this stream does produce an answer, it is usually correct.

Another possibility that has been explored in the literature is to favor answers that are returned by the highest number of streams. At TREC 2002, the LIMSI team experimented with an architecture consisting of two streams, one based on the local text collection and one based on the Web [4]. The underlying assumption is that similar answers coming from very different sources are more reliable, so the answer selection module favors those answer candidates found in the local collection, which have also been found in Web documents.

In the experiments on which we report below, we compare answer selection strategies that incorporate and build on ideas from both the MultiText and LIMSI approaches. Before reporting on those experiments, we briefly point to related work, not in question answering but in document retrieval, where the combination of retrieval runs is one of the recurring themes. It goes back at least to [7], and a recent overview of combination approaches is given in [3], describing combinations of document representation, query formulations, ranking algorithms, and search systems. The standard combination methods for merging ranked document lists are discussed and compared in [7, p.245/246].[1]

- *combMAX* Take the maximal query-document similarity score of the individual runs.
- *combMIN* Take the minimal similarity score of the individual runs.
- *combSUM* Take the sum of the similarity scores of the individual runs.
- *combANZ* Take the sum of the similarity scores of the individual runs, and divide by the number of non-zero entries.
- *combMNZ* Take the sum of the similarity scores of the individual runs, and multiply by the number of non-zero entries.
- *combMED* Take the median similarity score of the individual runs.

Since the similarity scores produced by different systems may differ radically, one often finds that a score normalization step takes place before one of the above methods is applied. In the question answering setting evaluations are not based on mean average precision (MAP) scores, but, in effect, on p@1. Because of this, there is no point in considering methods that may hurt early precision (but benefit MAP), such as combMIN, combANZ, and combMED. In our experiments in Section 5 we do consider answer selection methods based on the same intuitions as combMAX, combSUM, and, especially, combMNZ.

3 Overview of the Quartz Question Answering System

To make matters concrete, and to prepare for a report on our answer selection experiments, we will now zoom in on our own multi-stream open domain question answering, called Quartz. Quartz exists in two incarnation, a Dutch language version [12], and an English language version [11]. The brief overview below is based on the English version; we refer to [11] for more details.

After analyzing an incoming question, Quartz sends it to six streams in parallel, each of which is a small question answering system on its own. Every stream produces a ranked list of answer candidates (an answer pool). At the end of the process, the six answer pools are merged and the best candidate is returned as the final answer. While they share some components (named entity and part-of-speech taggers, parser, lexical resources, retrieval engine), the streams are independent and use very different strategies for answer extraction. The streams making up Quartz are briefly described as follows:

[1] In the setting of multi-lingual document retrieval, *round robin* is another merging method sometimes used [9]. Since we have to return a single final answer in the question answering setting, this method is not relevant for our discussions.

Table Lookup. This stream uses specialized knowledge bases constructed by preprocessing the collection. The stream exploits the fact that certain types of information (such as country capitals, abbreviations, and names of political leaders) tend to occur in a small number of fixed patterns. Similar to [6] we developed a small number of patterns for offline extraction of this information, using surface text and syntactic templates and WordNet. The knowledge base currently consists of 15 specialized tables. A fairly intricate knowledge base lookup process allows for non-exact matches.

Pattern Matching. Inspired by the success of methods based on pattern matching for certain question type [18], this stream exploits the fact that in some cases, the contextual format of an answer to a question can be back-generated from the question itself. For example, an answer to a question such as *2257. What is the richest country in the world?* may match the pattern `<Capitalized-Words>(,| is) the richest country in the world`. For each incoming question a set of possible answer patterns is generated and matches are attempted in relevant documents. We have two streams implementing this approach, *Web Patterns* and *Collection Patterns,* that use Google and our in-house IR engine, respectively, to retrieve relevant documents from the Web or from the AQUAINT corpus.

Ngram Mining. This stream, similar in spirit to [5], constructs a weighted list of queries for each question using a shallow reformulation process and then looks at word ngrams in the relevant retrieved document snippets. Quartz uses two variations of this stream: *Web Ngrams* and *Collection Ngrams,* using the Web and the local AQUAINT corpus, respectively.

Tequesta. Tequesta is a stream that implements a linguistically informed approach to QA. We refer to [11] for more details.

Each of the six streams described above provides a confidence score for each answer candidate. However, the actual values of the scores are calculated in a stream-specific way. The *Table Lookup*'s confidence depends on the number of occurrences of the relevant fact in the database and on the "exactness" of the match. The candidates' scores in the *Collection Patterns* and *Web Patterns* streams are based on the manually assigned accuracy of the patterns and the frequencies of the found answers. The *Web Ngrams* and *Collection Ngrams* streams use the number of ngram occurrences and some other features (e.g., presence of named entities of the appropriate type). Finally, the confidence scores of *Tequesta* stem from document scores given by the retrieval engine, the distance between question and answer terms in documents as well as a few additional syntactic, semantic and statistical features. Taking into account these differences, our answer selection module brings the candidates from all six streams together and selects the final answer.

Since Quartz's streams employ essentially different answering techniques, we expected that different streams would perform well on different question types. Indeed, an analysis of the results of the overall system and of the individual streams shows that each stream finds correct answers that are not found by other streams. Table 1 presents an evaluation of our best run at the TREC 2003 question answering track; we only consider factoid questions here, leaving out so-called list question and definition questions as these were assessed and scored differently. Evaluation is done by us, using the answer patterns provided by NIST [19].[2] We show the performance of our six streams for all questions and for questions of the 4 most frequent question types: *location* (e.g. *2316. What is the largest city in Austria?*), *number-many* (e.g. *1979. How many moons does Venus have?*), *date* (e.g. *1924. When was the first hair dryer made?*) and *person-ident* (e.g. *2301. What composer wrote "Die Götterdämmerung"?*). The row *alone* gives the number of questions correctly answered by the stream alone, the row *increase* gives the number of questions that the system would not answer without the stream (i.e., the number of questions correctly answered by the full system, but not answered by the system with this stream disabled).

Table 1. Comparison of the performance of the six question answering streams implemented in Quartz, on all question types and on the four most frequent question types.

		Correct answers						
Questions		*Table Lookup*	*Collection Patterns*	*Web Patterns*	*Collection Ngrams*	*Web Ngrams*	*Tequesta*	Total #q's
All	alone	71	39	51	39	65	63	413
	increase	17	1	6	3	30	30	
location	alone	16	9	2	9	17	15	67
	increase	2	1	1	3	10	10	
number-many	alone	5	5	5	5	10	17	46
	increase	0	0	0	0	2	9	
date	alone	8	3	8	3	7	14	37
	increase	1	0	2	0	4	3	
person-ident	alone	6	5	6	6	10	2	31
	increase	0	0	1	0	3	1	

The numbers in Table 1 clearly indicate that all of Quartz' six streams contribute to the system's performance, but the significance of the contribution depends on the question type. While each stream does find answer candidates

[2] Observe that these patterns do not distinguish between so-called *correct, inexact* and *unsupported* answers.

not found by any of the others (see the rows labeled *increase* in the table), many answers are found simultaneously by several streams, which explains the difference between *increase* and *alone* values.

4 Answer Selection in Quartz

In a single-stream question answering environment, the goal of answer selection is to choose from a pool of answer candidates the most likely answer for the question. As we pointed out in our discussion of related work, in a multi-stream environment, the task is complicated by the fact that the final answer now has to be selected from several pools of ranked candidates found by different streams. We need to compare answers coming from different sources and pick the one that is most likely to be correct. In this section we describe the way this has been implemented in Quartz.

4.1 Reranking and Score Normalization Using Web

As described before, all of Quartz' six streams produce pools of answer candidates together with numerical confidence scores. For the non-Web-based streams *(Table Lookup, Collection Ngrams, Collection Patterns* and *Tequesta)* we use Web hit counts (in a way similar to [13]) to adjust the stream's scores and thereby rerank the candidates within each pool. Apart from boosting correct answers this allows us to normalize the scores across the streams in order to make them comparable. However, this normalization does not take into account the fact that the initial scores are calculated differenly by each stream and have different and hard to predict ranges of possible values. The normalized candidate's score is based only on the stream's condifence and the Web frequency of the words of the question and of the answer.

4.2 Identifying Similar Answers

Next, we run a separate filtering module that mainly uses hand-coded heuristics to remove non-relevant candidates (e.g., it checks that answers to questions about person names indeed contain names, or for date questions the candidates bear temporal information).

In the next step, called *tiling,* the system tries to identify similar answers across the pools of answer candidates, using ideas similar to [8]. Two answers are considered similar if

- they are identical as strings, or
- the one is the substring of the other, or
- the edit distance between the strings is small compared to their length.

Note that according to this definition our notion of similarity is not an equivalence relation: the strings *"6th March 1863"* and *"May 1-3, 1863"* are both similar to *"1863"*, but not to each other. At the moment, when selecting one representative in a class of similar answers our system breaks ties randomly.

4.3 Weighting Streams

After similar answer candidates have been identified, the six pools must be merged and the answer with the highest confidence selected. However, even after adjusting scores with Web hit counts, the range of possible values is different for different streams, and it is likely that more adjustment is needed for scores to be comparable. We considered several options here:

- adjust each candidate's score by a factor w_{stream} that depends on the stream generating the candidate;
- adjust with a factor $w_{stream,qtype}$ that depends on both the generating stream and the type of the question being answered;
- use confidence values as given by the streams, without adjustments.

Intuitively, the weight w_{stream} serves two purposes: in addition to normalizing scores across streams it can indicate the reliability of the streams in general. Setting w_{stream} higher would favor answers coming from the particular stream. Similarly, using the $w_{stream,qtype}$ adjustment we can also indicate the trustworthiness of the streams for different question types. The last option (no adjustment) corresponds to the (not unreasonable) belief that the scores of different streams are comparable "as is" since we have already used Web hit counts which might also have normalizing effect.

There are several ways to choose actual values for the stream weights. They can be made equal, assigned manually based on an analysis of the performance of the streams, or they can be learned automatically, from a training set of question/answer pairs. While the first two approaches are fairly straightforward and leave little room for variation, the *learning* of weights can be done in a variety of ways. Here, we describe a naive learning algorithm that proceeds by iteratively adjusting weights for each stream-question type pair. This is the method used in the experiments of Section 5.

Given a set of questions and correct answers, we learn the weights for the streams so as to maximize the number of questions answered correctly by the system as a whole. The algorithm starts with initial weights for all pairs (stream, question type):

$$w_{s,qt} := \frac{\#\text{ correct answers by stream } s \text{ to questions of type } qt}{\sum_{s'} \#\text{ correct answers by stream } s' \text{ to questions of type } qt}.$$

Then, for each question q of type qt, if the stream s^* found a correct answer a^*, but with the current stream weights the answer selection module chooses an incorrect answer, the weight $w_{s^*,qt}$ is increased so that the correct answer is selected. Let $score_s(q,a)$ denote the confidence score assigned by the stream s to an answer candidate a for a question q. Then the weight of the stream s^* is adjusted as

$$w_{s^*,qt} := w_{s^*,qt} \cdot \frac{\max_{s,a} score_s(q,a) \cdot w_{s,qt}}{score_{s^*}(q,a^*) \cdot w_{s^*,qt}}.$$

The weights of the other streams remain the same. Then all the weights are normalized so that $\sum_s w_{s,qt} = 1$. It is easy to see that after this adjustment the answer a^* will have the highest weighted score and thus will be selected as the final answer. This procedure is repeated for all questions and then several times for the whole training set. Although this algorithm does not find globally optimal weights and even does not necessarily converge (we chose the number of iterations empirically, so that it gives the best performance on the set of training questions), our experiments showed that it does increase substantially the number of correct answers produced by the system.

Clearly, standard, more sophisticated and better understood machine learning techniques could also be applied to the task of learning optimal weights. However, one of our aims was to see whether the idea of learning stream/question type weights could be made to work in a straightforward way; it appeared to be easier to implement the naive and intuitively clear algorithm outlined above rather than squeeze the task into any of the well-known but more complicated methods. While our simple approach gives encouraging performance improvement, in future work we plan to investigate other, classical techniques.

4.4 Creating the Final Pool

Once the confidence scores of all answer candidates have been adjusted, there are still a number of ways to create a single pool of candidates:

- similar answers (as identified during tiling) are merged and their confidence scores added;
- similar answers are merged and those answers are favored that come from a larger number of streams.

As pointed in Section 2, the second approach was used in [4] for a question answering system consisting of two streams: collection- and Web-based. The underlying assumption was that similar answers coming from very different sources are more reliable. The extension of the technique to many sources may allow us to use in full the redundancy of the multi-stream architecture.

5 Experiments

Our aim was to systematically evaluate the effect of different options for answer selection in our heterogeneous QA system. Here is a short summary of the options we considered:

- use tiling or not;
- use weights based on stream (w_s) or based on stream and question type ($w_{s,qt}$);
- the choice of weights: equal, manually assigned or automatically learned from past experience;
- exploit or not the redundancy in full: consider only those answer candidates that come from the greatest number of streams, and among those pick the one with the highest final score.

For the purposes of our experiments we also created stream/question type weights manually. We analyzed the candidates of Quartz's streams for the 500 TREC 2002 questions and assigned a confidence value (0, 2, 5 or 10) to each stream and question type, based on an intuitive understanding of how good the candidates were. Observe that since these 500 questions also constituted more than a half of our training/testing collection (see below), the results for the manual voting on which we report below might be an over-estimate.

For a proper evaluation of different answer selection schemes we took the set of factoid questions from TREC 2002 and 2003 question answering tracks (913 questions) together with the patterns of correct answers provided by NIST. Since one of the options involves learning and, moreover, since our aim was to understand the *significance* of the differences, we randomly split the question set into training and evaluation sections (180 question for evaluation, the rest for training) and repeated the experiments 50 times with different splits. The splits of the question sets had to satisfy the following constraint: for all question types 80% of the questions are in the training set.

We evaluated the following variants of the answer selection module:

ET equal weights, tiling
MT manual weights, tiling
AT automatically learned weights, tiling
A automatically learned weights, no tiling
ATU automatically learned weights with weights not dependent on question type, tiling
ATB automatically learned weights, only answers from the largest number of streams are left, tiling
ETB equal weights, only answers from the largest number of streams are left, tiling.

We used a one-tailed t-test to establish the statistical significance of the differences observed in our experiments.

6 Results

Table 2 shows the evaluation results for the seven answer selection schemes: the average number of correct answers on a set of 180 randomly chosen questions after training on the remaining 733, measured after 50 iterations. It also shows differences between several voting schemes.

Not surprisingly, the answer selection scheme using stream/question type weights, tiling and stream redundancy (ATB) improves significantly over the simple baseline scheme (ET: with tiling but no weights) and it allows our system to give over 30% more correct answers. Moreover, both stream weighting (AT over ET) and exploiting the system's redundancy (ETB over ET) alone make significant improvements as well.

Tiling gives better performance (AT over A), although it sometimes results in answers that would have been judged *inexact* rather than *correct* by human

Table 2. Comparison of answer selection methods, measuring the average number of correct answers out of 180 randomly chosen questions (50 iterations).

Baseline		Modifications			
Scheme	#answers	Scheme	#answers	%change	Significance
ET	39.9	MT	52.0	+30.3%	** ($p < 0.001$)
		AT	50.8	+27.3%	** ($p < 0.001$)
		ATB	**52.6**	**+31.8%**	** ($p < 0.001$)
A	43.4	AT	50.8	+17.1%	** ($p < 0.001$)
ATU	48.2	AT	50.8	+5.4%	** ($p < 0.001$)
AT	50.8	ATB	52.6	+3.5%	* ($p < 0.05$)
ET	39.9	ETB	48.8	+22.3%	** ($p < 0.001$)

assessors. E.g., for question *1912. In which city is the River Seine?* our system returns the answer *Paris Opera House* rather than *Paris* only because of the tiling. In the TREC 2003 QA track over 25% of the "not wrong" answers produced by our system were judged *inexact.* While tiling does indeed help to boost strings containing correct answers, more informed filtering and type checking need to be done in the end.

The difference between AT and ATU (weighting depends on question type or not) indicates that it does indeed make sense to look at the type of the question during final answer selection, rather than simply basing the decision on the overall performance of the streams. This is explained by the fact that the streams show very different accuracy on different question types.

It is interesting to note that there is no significant difference between selection schemes with manually assigned weights (MT) and with weights chosen so as to optimize the performance on a training set of questions (AT). This is an important point as it demonstrates that there is no need to manually analyze the performance of the streams on different questions, which is a very laborious process given the number of streams and different question types. Even a very naive learning method can produce a set of weights that yields an end-to-end performance of the question answering system that equals the performance obtained with manual assignment of weights.

7 Conclusions

We have described experiments with different methods for answer selection in a multi-stream question answering system. We examined the impact of local tiling techniques, assignment of weights to streams based on past performance and/or question type, as well redundancy-based ideas. Our main finding is that redundancy-based ideas in combination with naively learned stream weights conditioned on question type work best, and improve significantly over baselines. In future work we plan to apply other, better understood machine learning techniques to the task of learning optimal weights, and use more informed and reliable methods for answer tiling and answer filtering.

Summing up, open domain question answering systems are becoming more and more complex, increasingly relying on multiple approaches and multiple external resources. Since we do not make any assumptions about the nature of the streams, or even about the types being used in question classification, we believe that the methods and results that we have presented are widely applicable in open domain question answering.

Acknowledgments. We want to thank Gilad Mishne for help and advice, and we are grateful to our referees for useful comments. This research was supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. Maarten de Rijke was also supported by NWO under project numbers 612-13-001, 365-20-005, 612.069.006, 612.000.106, 612.000.207, and 612.066.302.

References

1. J. Chu-Carrol, J. Prager, C. Welty, K. Czuba, and D. Ferrucci. A multi-strategy and multi-source approach to question answering. In *Proceedings of TREC 2002,* pages 281–288, 2003.
2. C.L.A. Clarke, G.V. Cormack, G. Kemkes, M. Laszlo, T.R. Lynam, E.L. Terra, and P.L. Tilker. Statistical selection of exact answers. In *Proceedings of TREC 2002,* pages 823–831, 2003.
3. W.B. Croft. Combining approaches to information retrieval. In W.B. Croft, editor, *Advances in Information Retrieval,* pages 1–36. Kluwer Academic Publishers, 2000.
4. G. de Chalendar, T. Dalmas, F. Elkateb-Gara, O. Ferret, M. Hurault-Plantet B. Grau, G. Illouz, L. Monceaux, I. Robba, and A. Vilnat. The Question Answering System QALC at LIMSI, Experiments in Using Web and WordNet. In *Proceedings of TREC 2002,* pages 407–416, 2003.
5. M. Banko et al. AskMSR: Question answering using the Worldwide Web. In *Proc. EMNLP 2002,* 2002.
6. M. Fleischman, E.H. Hovy, and A. Echihabi. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings ACL 2003,* pages 1–7, 2003.
7. E.A. Fox and J.A. Shaw. Combination of multiple searches. In *Proceedings TREC-2,* pages 243–252, 1994.
8. M. Greenwood, I. Roberts, and R. Gaizauskas. The University of Sheffield TREC 2002 Q&A system. In *Proceedings of TREC 2002,* pages 823–831, 2003.
9. D. Hiemstra, W. Kraaij, R. Pohlmann, and T. Westerveld. Translation resources, merging strategies, and relevance feedback for cross-language information retrieval. In *Proceedings CLEF 2000,* pages 102–115. Springer, 2001.
10. E. Hovy, H. Hermjakob, M. Junk, and C.-Y. Lin. Question answering in Webclopedia. In *Proceedings TREC-9,* 2000.
11. V. Jijkoun, J. Kamps, G. Mishne, C. Monz, M. de Rijke, S. Schlobach, and O. Tsur. The University of Amsterdam at TREC 2003. In *TREC 2003 Notebook Papers,* 2003.
12. V. Jijkoun, G. Mishne, and M. de Rijke. How frogs built the Berlin Wall. In *Proceedings CLEF 2003.* Springer, to appear.

13. B. Magnini, M. Negri, R. Prevete, and H. Tanev. Is it the right answer? Exploiting web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL),* pages 425–432, 2002.
14. D. Moldovan, S. Harabagiu, M. Paşca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. The structure and performance of an open domain question answering system. In *Proceedings ACL 2000,* pages 563–570, 2000.
15. D.Moldovan, M. Paşca, S. Harabagiu, and M. Surdeanu. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems,* 21:133–154, 2003.
16. C. Monz and M. de Rijke. Tequesta: The University of Amsterdam's textual question answering system. In *Proceedings TREC 2001,* pages 519–528, 2002.
17. J. Prager, E. Brown, A. Coden, and D. Radev. Question-answering by predicitive annotation. In *Proceedings SIGIR 2000,* pages 184–191, 2000.
18. M.M. Soubbotin and S.M. Soubbotin. Use of patterns for detection of likely answer strings: A systematic approach. In *Proceedings TREC 2002,* pages 325–331, 2003.
19. Text REtreival Conference (TREC). URL: `http://trec.nist.gov`.
20. J. Xu, A. Licuanan, J. May, S. Miller, and R. Weischedel. TREC 2002 QA at BBN: Answer Selection and Confidence Estimation. In *Proceedings of TREC 2002,* pages 96–101, 2003.

A Bidimensional View of Documents for Text Categorisation

Giorgio Maria Di Nunzio

Department of Information Engineering,
University of Padua.
dinunzio@dei.unipd.it
http://www.dei.unipd.it/~dinunzio/

Abstract. The question addressed in this paper is to find a bidimensional representation of textual documents for the problem of text categorisation. The projection of documents is performed following subsequent steps. The main idea is to consider a possible double aspect of the importance of a word: the local importance in a category, and the global importance in the rest of the categories. This information is combined properly and summarized in two coordinates. Then, a machine learning method may be used in this simple bidimensional space to classify the documents. The results that can be obtained in this space are satisfactory with respect to the best state-of-the-art performances.

1 Introduction

The statistical approach to the problem of Text Categorisation (TC) has gained over the last years the interest of the researchers [1,2]. It is a very active area of research mostly due to the needs of organising large amount of documents available especially online. Recent works show that it is possible to achieve a good tradeoff between efficiency and effectiveness through statistical models, such as Naive Bayes [3,4], ridge logistic regression [5], and support vector machines [6]. For an exhaustive introduction to the problem of Text Categorisation we suggest [7] and [8].

One of the most important challenges that this field faces is the scale of the space of words and documents, they easily may reach tens or hundreds of thousands. This intrinsic high dimensionality affects dramatically the cost both from a computational point of view and from the intensive memory usage, for managing such a number of items.

In order to reduce the amount of objects to deal with, a *word selection* (usually called *feature selection*) is widely performed. This selection tries to reduce the space of the terms by means of a function that measures the importance of some particular term [9,10]. Other approaches which reduce this high dimensional space are document clustering methods such as: projection based method like latent semantic analysis [11,12], self-organizing maps [13] and multidimensional scaling [14].

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 112–126, 2004.

Following the second approach, an interesting problem would be that of automatically generating document clusters onto a bidimensional space in order to be at the same time viewable and manageable. With this idea in mind a new word-weighting scheme would be necessary, leaving aside the old, but still widely used, *tf*×*idf* weighting scheme [15]. One of the rare recent works that seriously questions the problem of studying new weights is [16].

In this paper we propose a technique to project documents into a bidimensional plane using few easy understandable parameters that capture the information present in the original high dimensional space. Within this new space a heuristic learning method is proposed, trying to exploit the geometrical clues given by the graphical representation offered by this bidimensional view.

The paper is organised as follows. Section 2 presents the formal framework of the approach proposed. In Sect. 3 a heuristic algorithm (here, it will be simply referred to as *Heuristic*) for categorising documents is proposed. In Sect. 4 the results of experimenting with this representation on Reuters-21578 benchmark[1] are described. In Sect. 5 our future perspectives are claimed. In Sect. 6 some final remarks are made.

2 Formal Framework

The construction of an automatic text classifier usually needs an initial corpus Ω of pre-classified documents under some predefined categories. In a supervised learning approach this initial corpus is split into two subsets: the *training* set *Tr* and the *test* set *Te,* where $Te = \Omega$ - *Tr*. The former is used to train the system and the latter is used to measure the effectiveness of the same.

When a *k-fold validation* approach is followed, the training set is furtherly split into k subsets equally sized. A hypothesis is produced by training on $k -$ 1 subsets and the testing is done on the remaining subset. This is repeated iteratively k times, and the observed errors are averaged to form the k-fold estimate. The approach used in this paper is: to take the whole training set to estimate the parameters for the projection of documents (explained in this Sect.), and to use the *k-fold* procedure to train the classifier on the bidimensional representation (see the details in Sect. 3).

The whole idea of document projection lies on a very simple intuition. Given a set of categories *C,* a word may show two different aspects: its importance in a particular category (the category of interest), and its importance in the other categories. We will sometimes use the terms *local* or *global* according to which aspect we are focusing on.

To begin with, the two parameters Presence and Expressiveness will be defined; they actually represent the basic bricks the model is built with. Then, a *local* or *global* weight is presented using Presence and Expressiveness, and a *local* and *global* energy of a category is defined in order to summarize all the information found. Finally, the last step shows how to project a document on a Cartesian plane.

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

In order to have a coherent symbolism among the formulas, some general definitions are given here: we shall assume a set of predefined categories $C = \{c_1, c_2, ..., c_i, ..., c_n\}$, and shall indicate with $d_{j,i}$ the j-th document that belongs to the i-th category (e.g. $c_1 = \{d_{1,1}, d_{2,1}, ..., d_{j,1}, ..., d_{N_i,1}\}$), having a generic category c_i a number of documents N_i. The set of distinct terms of a category c_i is T_i, while a generic term of the *vocabulary* will be $t \in \bigcup_{i=1}^{n} T_i$. Moreover, the category of interest will be usually indicated as c_i, and the other categories as the *rest of the world* (or *ROTW*) which is equal to the difference set $C - c_i$. In this way, given a category c_i the *ROTW* is uniquely defined. The words: *word, term* and *feature* are used in this paper as synonyms.

2.1 Presence and Expressiveness

Presence and Expressiveness are the basic parameters on which all the model is built and their formal definition is presented in this section (see also [17]). An implicit assumption is made, which is somewhat a relaxation of one of the *monotonicity assumption* of term weighting in Information Retrieval [16,18]: the frequency of words inside a document is not important. In other words, what is relevant is the fact that the word does or does not appear in the document. This assumption has been made because the benchmark used is a collection of short news agency articles, but in general, it might be not valid.

Presence P(t,c) : it is an intra-category measure, in the sense that it is calculated for each category. For each term t, $\hat{P}(t, c_i)$ is estimated as the percentage of documents of the category c_i in which the term t appears:

$$\hat{P}(t, c_i) = \frac{N_{i|t}}{N_i} \qquad 0 \leq \hat{P}(t, c_i) \leq 1 \ , \tag{1}$$

where $N_i = \sum_{j=1}^{m} d_{j,i}$ is the number of documents of c_i, and $N_{i|t} = \sum_{j=1}^{m} (d_{j,i}|t)$ is the number of documents in which the term t is present. It is worth noting that $\hat{P}(t, c_i)$ is equal to zero for all the terms $t \notin T_i$.

Expressiveness E(t,c) : it is an inter-category measure, and it exploits the information given by the Presence. In particular, the Expressiveness of a term t in a category c_i estimates how much t does *not* appear, on average, in the *ROTW:*

$$\hat{E}(t, c_i) = 1 - \frac{\sum_{k=1}^{n} \hat{P}(t, k)}{n-1} \qquad k \neq i \qquad 0 \leq \hat{E}(t, c_i) \leq 1 \ . \tag{2}$$

What the Expressiveness does is: given a category c_i, calculate the mean of the Presence of a term t in the rest of the world, and subtract it from one. Note that this subtraction is done in order to match the numerical value with the meaning of the word *expressiveness:* the more a term t is representative for a category (it is expressive) the higher the value of $\hat{E}(t, c_i)$ is.

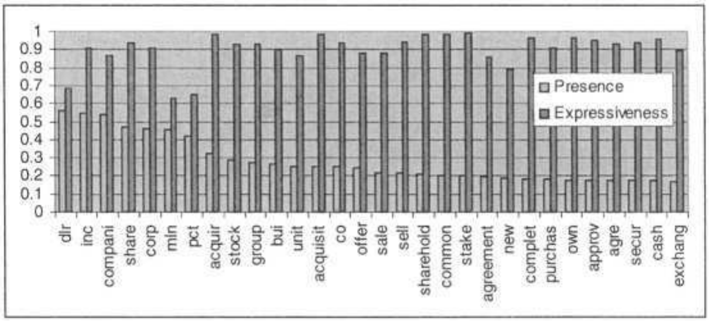


Fig. 1. Presence and Expressiveness of the first thirty terms of a category, ordered by Presence. The category is *Acquisitions* from the Reuters-21578 benchmark

One of the first important results that can be achieved by Presence and Expressiveness is the straightforward graphical impact. In Fig. 1 the values of Presence and Expressiveness for the first thirty terms ordered by Presence are shown. They have been taken from category *Acquisitions* of the Reuters-21578 benchmark.

2.2 Local and Global Term Weighting

In this section a weighting scheme that reflects the twofold aspect of a word discussed above is proposed. In particular, the problem of how to weight a word will be seen from two points of view:

1. how to compute the *local importance* of a term in a category c_i, with respect to the *ROTW;*
2. how to compute the *global importance* of the same term in the *ROTW* with respect to the c_i.

The first point is more intuitive; the local weight of a term may be defined as the product of Presence and Expressiveness:

$$localW(t, c_i) = \hat{P}(t, c_i) \cdot \hat{E}(t, c_i) \ . \qquad (3)$$

localW may be interpreted in the following manner: the Presence of a term is penalized by a factor proportional to its Expressiveness. Figure 2 shows the result of the local weight $\hat{P}(t, c) \times \hat{E}(t, c)$ for the same thirty terms of Fig. 1. For example (see Fig. 1), the word *dlr* has a Presence greater than the word *corp* (0.56 against 0.47), nevertheless (see Fig. 2) the Expressiveness of *dlr* penalizes the local weight more than *corp*. The result is that *corp* has a *localW* greater than *dlr*. The second point requires a small effort more. Let's state the dual problem: consider the rest of the world as our category of interest, and the category c_i as the new *ROTW*. The new Presence would be the mean of the presence of the term in the *ROTW,* that can be rewritten as:

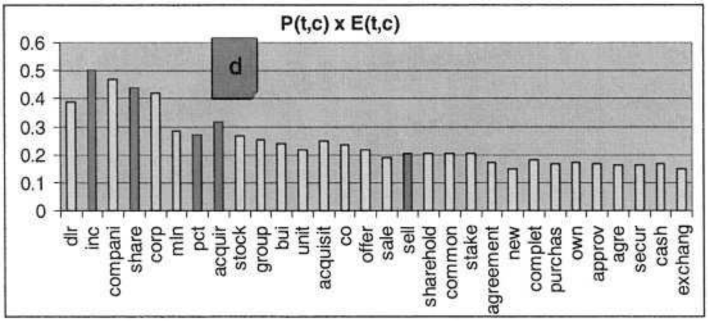


Fig. 2. Here the same thirty words of Fig.(1) are shown. Now, the y axis represents the product $\hat{P}(t,c_i) \cdot \hat{E}(t,c_i)$. The words that appear in a generic document d are highlighted. The projection of the document is shown in Fig. 3 as $d(0.24, 0.13)$

$$\frac{\sum_{k=1}^{n} \hat{P}(t,c_k)}{n-1} = 1 - \left(1 - \frac{\sum_{k=1}^{n} \hat{P}(t,c_k)}{n-1}\right) = 1 - \hat{E}(t,c_i) \quad k \neq i \ . \tag{4}$$

Given this new point of view, the *globalW* of a word is defined as:

$$globalW(t,c_i) = (1 - \hat{E}(t,c_i)) \cdot (1 - \hat{P}(t,c_i)) \ , \tag{5}$$

where the second factor $(1 - \hat{P}(t,c_i))$ is the definition of the Expressiveness, accordingly to the dual problem.

2.3 Local and Global Energy of a Category

Once the weights *localW* and the *globalW,* are available for each term, a measure that summarizes the twofold information is needed. It is called *energy* and, once again, it may be seen from two points of view:

1. compute the energy of the category c_i of interest with respect to the rest of the world, the *local energy;*
2. compute the energy of the rest of the world with respect to the category c_i of interest, the *global energy.*

Using (3) and (5) the local energy function $\tilde{P}$ is defined as the sum of all the local weights *local W:*

$$\tilde{P}_i = \sum_t \hat{P}(t,c_i) \cdot \hat{E}(t,c_i) = \sum_t localW(t,c_i) \ , \tag{6}$$

while the global energy function $\tilde{E}$ is defined as the sum of all the global weights *globalW.*

$$\tilde{E}_i = \sum_t (1 - \hat{E}(t,c_i)) \cdot (1 - \hat{P}(t,c_i)) = \sum_t globalW(t,c_i) \ . \tag{7}$$

Note that the local energy of a category (as well as the global one) will trivially vary from category to category according to the number of words that appear in a category, but also according to how much a topic of a category is specific or general. An empirical investigation showed that categories with few words that have a high *localW*, may have a local energy higher than categories populated with lots of words but with a low *localW*.

2.4 Cartesian Representation of Documents

The final aim of the bidimensional representation of the documents can be reached with one last step. The idea is to re-use the concept of energy also for the documents, that is to say:

1. what is the energy of a document d in the category of interest c_i, and what is the ratio between this energy and the overall $\tilde{P}$?
2. what is the energy of a document d in the *ROTW*, given c_i, and what is the ratio between this energy and the overall $\tilde{E}$?

Denoting a generic term that appears in a document d as $\dot{t}$, the coordinate X_i of a document answers to the first point, it is defined as:

$$X_i = \frac{\sum_{\dot{t} \in d} \hat{P}(\dot{t}, c_i) \cdot \hat{E}(\dot{t}, c_i)}{\tilde{P}_i} , \tag{8}$$

where the energy of the document d in the category c_i is computed by the sum $\sum_{\dot{t} \in d} \hat{P}(\dot{t}, c_i) \cdot \hat{E}(\dot{t}, c_i)$. Accordingly, the second point is answered by the Y_i coordinate of document d:

$$Y_i = \frac{\sum_{\dot{t} \in d} (1 - \hat{E}(\dot{t}, c_i)) \cdot (1 - \hat{P}(\dot{t}, c_i))}{\tilde{E}_i} . \tag{9}$$

Here, the energy produced by d in the *ROTW* is the sum $\sum_{\dot{t} \in d} (1 - \hat{E}(\dot{t}, c_i)) \cdot (1 - \hat{P}(\dot{t}, c_i))$.

Both X_i and Y_i are defined from 0, when d does not contain any term of category c_i (or any term of the rest of the world), to 1 when d contains all the terms of the category (or all the terms of the *ROTW*). Usually, a typical document that contains only some tens of words can never obtain a X_i or a Y_i close to 1.

Figure 2 shows a document d whose words belonging to the category c_i are highlighted. To give a numerical example of the use of X_i and Y_i, suppose that the thirty words in the figure are indeed the set of terms of category c_i. The energy of this category would be:

$$\tilde{P}_i = \sum_t \hat{P}(t, c_i) \cdot \hat{E}(t, c_i) \simeq 7.04 .$$

If this was the energy, the X_i coordinate for the document would be:

$$X_i = \frac{\sum_{\dot{t} \in d} \hat{P}(\dot{t}, i) \cdot \hat{E}(\dot{t}, i)}{\tilde{P}_i} \simeq \frac{1.73}{7.04} \sim 0.24 .$$

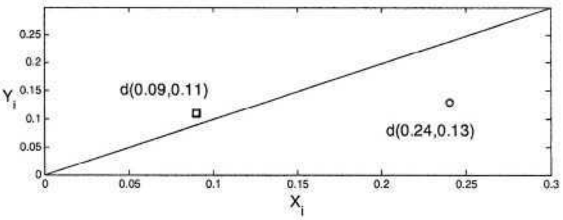


Fig. 3. Two documents as points on the Cartesian plane. The solid line represents the points where $X_i = Y_i$. The documents below this line should be the ones that belong to the category of interest

Following the same reasoning for the calculation of coordinate Y_i, not shown here for the sake of brevity, a pair which represents a point in the Cartesian space is obtained:

$$d(X_i, Y_i) = d(0.24, 0.13) \ .$$

The result of all these passages is shown in Fig. 3, a Cartesian representation of a document. Two documents are plotted: $d(0.24, 0.13)$ (the circle one, the one just found with the numerical example) and $d(0.09, 0.11)$ (the square one). The solid line represents the points in the space in which $X_i = Y_i$, that is when a document d *produces* the same energy both in category c_i and in the rest of the world. The interpretation of the position of the documents in the space is given in the next Sect. It is worth noting that n different projections (in n different spaces) of a document d exist, depending on the category of interest under investigation.

3 Categorising Documents with a Heuristic Approach

Given the new bidimensional representation of a document in the space of the category of interest c_i, a method for deciding whether a document belongs under c_i or not is needed. An intuitive idea is: since X_i and Y_i are measuring how much energy the document possesses in the category c_i or in the *ROTW,* it is plausible to imagine that when X_i is greater than Y_i, the document belongs to the category, otherwise it does not.

Figure 3 already showed an example with two documents. Bearing in mind the intuitive idea, it would mean to accept the document $d(0.24, 0.13)$ and reject the other $d(0.09, 0.11)$.

In the following sections it will be shown that, in the real case, this crisp separation does not exist. Anyway, the distribution of the documents belonging to c_i or to the *ROTW* will present some regularity, such that a statistical two-class separation method might be used, or even a heuristic approach.

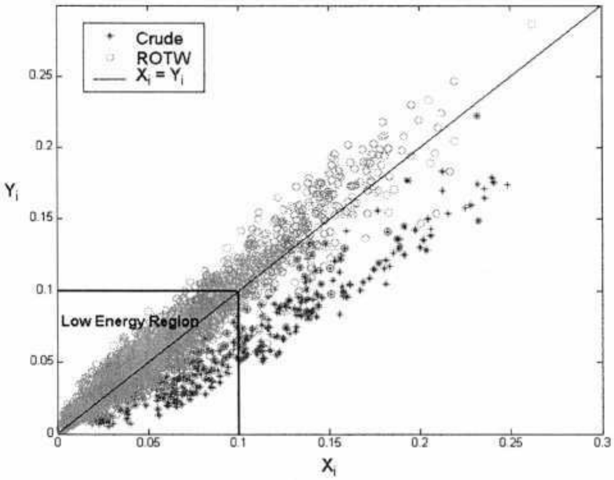


Fig. 4. Graphical representation of a collection of documents. The training documents of Reuters-21578 are projected into the space of category *Crude*. The stars are the documents that belong to the category of interest, the circles are the *ROTW* documents. In the lower-left region, a *Low Energy Region* is shown, where many *ROTW* documents fall below the equality line $X_i = Y_i$

3.1 The Real Distribution of Documents

Let us suppose to have a collection of documents, the Reuters-21578 described in Sect. 4. In Fig. 4 an example of projection of documents in the space of a category named *Crude* is shown. The documents that belong to this category are marked with a star, the rest of the world documents are marked with a circle, the solid line represents again the points where $X_i = Y_i$ (also referred as *equality line*).

Some considerations about this real-world case may be immediately drawn:

- almost all the documents of the category of interest are below the equality line (more than 99%), and this seems to be a general result, as it happens also for the other categories;
- there is a certain number of documents of the rest of the world that fall below the equality line; in this particular case almost the 23% of documents of the *ROTW* fall below this line, but this number may vary from 10% to 50%, which is a great amount of documents that lower the performance of the categorisation system;
- most of these badly categorised documents fall in the lower-left region of the plane, which means the region of documents that produce little energy.

These considerations show that the intuitive idea is not effective in general.

3.2 A Heuristic Approach versus SVM

To solve this two-class separation problem, two different learning methods are used: a *Heuristic* learning method that captures some geometrical clues given by the bidimensional representation and the support vector machines (SVM).

The *Heuristic* approach can be stated as follows: consider the angular region whose vertex is the point (0, q), bounded by the semi-lines $Y_{i,c} = m_c \cdot X_i + q$ and $Y_{i,r} = m_r \cdot X_i + q$, where $Y_{i,c}$ and $Y_{i,r}$ are the interpolating lines of the set of documents of c_i and the set of documents of *ROTW,* respectively and both constrained to pass through the point (0,q) (see Fig. 5). Within this region (called *Optimal Region*), the optimal separating line should be found. The equation of the optimal line would be:

$$Y_{i,opt} = m_{opt} \cdot X_i + q \qquad m_c \leq m_{opt} \leq m_r \ .$$

The cost of finding the best solution will depend on how well the interval m_c to $m_r$divided, and how far the point (0, q) is chosen. In our experimentation, the interval of the angular coefficient was equally divided into one hundred parts, while the coordinate q was bounded from 0.00 to –0.03 with intervals of 0.001.

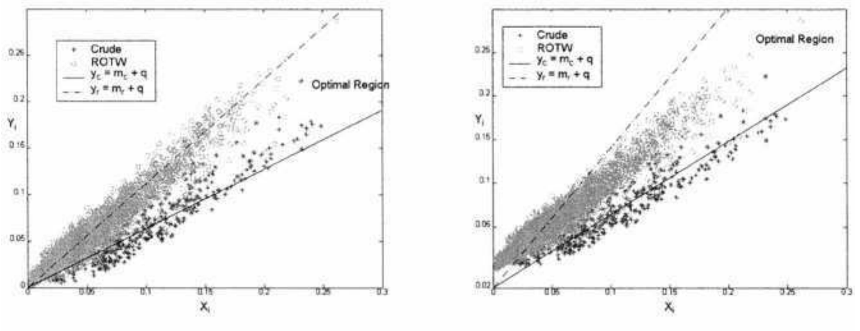
SVM is a statistical learning method that attempts to learn a hyperplane in a high-dimensional space (usually the space of terms) that separates the positive samples from the negative ones with the maximum possible margin [16,6,19]. The SVM^{Light} package[2], an implementation of SVM, is employed with the default parameters, that means a linear kernel is used together with an additional parameter '$-b$' in order to have a biased hyperplane.

4 Experimental Setup

To make the evaluation of the Category Energy Model comparable to most of the published results on Text Categorisation (TC), the Reuters-21578 corpus was chosen as a benchmark. During the last years this corpus has been used as a standard benchmark on which many TC methods have been evaluated. The experimentation in this paper used the ModApte split of Reuters-21578 where 75% of the stories (9603) are used as training documents to build the classifier and the remaining 25% (3299) to test it. Only the 10 most frequent of the 135 potential categories herein are used; these 10 categories account for almost the 75% of the training instances, while the remainder is distributed among the other 125. Some text preprocessing was done on the documents: a first cleaning was done removing all the punctuation marks and all the numbers and converting all the letters to lowercase. A stoplist of 232 words and contractions (i.e. 're, don't, etc.) was used to remove the most frequent words of the English language, finally the English Porter stemmer[3] was used as the only method to reduce the space of terms.

[2] Freely available at `http://svmlight.joachims.org/`

[3] Freely available at `http://www.tartarus.org/~{}martin/PorterStemmer/`



(a) q = 0

(b) q = -0.02

Fig. 5. The *Heuristic* method finds the optimal separating line $Y_{i,opt} = m_{opt} \cdot X_i + q$ in the angular region (called Optimal Region). The region is bounded by the two interpolating lines $Y_{i,c}$ and $Y_{i,r}$ and the vertex (0, q). In the two sub-figures the training documents of Reuters-21578 are mapped into the category *Crude*. On the left, Fig.(5(a)), the lines are constrained by the parameter q = 0. On the right, Fig.(5(b)), they are constrained by q = –0.02

4.1 Effectiveness Measures

In order to evaluate the accuracy of the classifier the standard IR measures have been computed. Recall ρ_i and Precision π_i are defined for each category c_i as (using the same notation of [8]):

$$\rho_i = \frac{TP_i}{TP_i + FN_i} \;,\quad \pi_i = \frac{TP_i}{TP_i + FP_i} \;,$$

where TP_i *(true positive)* is the number of documents correctly classified under category c_i, and FN_i *(false negative)* and FP_i *(false positive)* are defined accordingly. The performance of the classifier for the whole set of categories was estimated according to both the two methods of *microaveraging* and *macroaveraging* as shown in Table 1.

Finally, the F_β function (see [20]) was calculated, and in particular when $\beta = 1$, which is the most used value:

$$F_\beta = \frac{(\beta^2 + 1) \cdot \pi \cdot \rho}{\beta^2 \cdot \pi + \rho} \;,\qquad F_1 = \frac{2 \cdot \pi \cdot \rho}{\pi + \rho} \;. \tag{10}$$

4.2 Analysis of the Results

The results reported in the first two columns of Table 2, and compared with four state-of-the-art methods from the literature, are encouraging. In fact, although

Table 1. Averaged effectiveness measures

	micro-averaging	macro-averaging
recall	$\rho^{\mu} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n}(TP_i + FN_i)}$	$\rho^{M} = \frac{\sum_{i=1}^{n} \rho_i}{n}$
precision	$\pi^{\mu} = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n}(TP_i + FP_i)}$	$\pi^{M} = \frac{\sum_{i=1}^{n} \pi_i}{n}$

the performances are a little bit lower than the state-of-the-art ones, the gain in understandability of the parameters, in impact of this graphical representation and in low computational costs open new prospectives in this direction.

For the *Heuristic* method the *k-fold validation* approach has been adopted and the optimal final solution is found averaging the parameter m_{opt} and q with the number k of tests. The best result has been obtained with $k = 5$, although it does not seem that the number of subsets k influence dramatically the results (it has been experimented with a number of k from 1 to 10 and the average of the results for the macro-precision and micro-precision were 79.9 and 86.8 respectively). The results of the *Heuristic* are reasonably good and close to the best performances of the learning methods presented in the table. This approach may be considered almost equal to the Multinomial model presented in [3] (0.6 less both for the macro-average and the micro-average), while it performs a bit worse with respect to the others (from ~ 3.0 up to ~ 6.0 less).

Another important consideration is that SVM^{Light} does not perform as well as expected and it always performs worst with respect to the *Heuristic* approach. The question is still open and needs to be investigated thoroughly.

An important issue comes out from the row-by-row investigation of Table 2 together with the graphs of the categories. There are three categories where the *Heuristic* performs badly with respect to the other methods, they are: Acquisitions, Wheat and Corn. Figure 6 shows the projections of the Reuters-21578 documents in the space of category Acquisitions. The stars are the documents belonging to this category and the circles are the *ROTW* documents. In the low energy region there is an area of documents of the *ROTW* that clearly overlap the ones of the category of interest. In Fig. 7, only the documents of Acquisitions, Earnings and Corn are shown. From this figure it can be seen that the overlapping region is mostly populated by the category Earnings. While the Corn category (for the sake of clarity it could not be possible to show all the categories, but the same happens with the others) is quite far from the category of interest. The same happens for the Wheat and Corn that overlap with the Grain category.

Table 2. The best F–1 Performance results for the ModApte version of Reuters-21578 (top 10 classes) obtained with our method compared with four different recent works. The first two columns are the results of the *Heuristic* approach (with k=5 for the *k-fold validation*) and the SVM^{Light} package. The third column (multinomial) is reported by [3] (only the averaged measures are available). The fourth column (logistic regression) is reported by [3] (and previously published by [5]). The fifth column (Logistic Gaussian) is reported by [21]. The sixth column (support vector machines) is reported by [16](only the averaged measures are available)

	Heuristic	SVM^{Light}	Multi	Log Reg	Log Gauss	SVM
earn	94.9	91.3		98.4	98.1	
acq	85.8	83.1		95.2	95.3	
money-fx	76.0	70.8		75.2	74.4	
grain	91.7	72.1		84.1	85.9	
crude	83.5	76.2		85.9	84.8	
trade	75.2	71.3		72.9	73.4	
interest	75.0	64.0		78.2	75.9	
wheat	78.9	54.6		88.2	88.9	
ship	84.9	72.2		81.9	82.4	
corn	56.3	44.2		88.7	86.2	
macro-averaging	**80.4**	**70.0**	**81.0**	**85.3**	**83.7**	**85.6**
micro-averaging	**87.1**	**82.1**	**87.7**	**91.4**	**90.7**	**92.1**

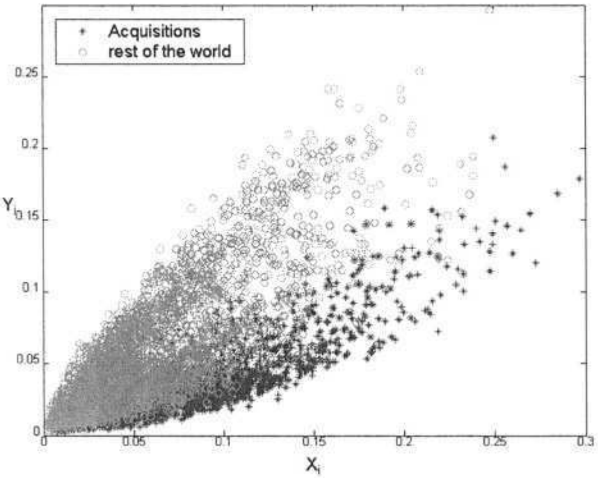


Fig. 6. The training documents of the Reuters-21578 are plotted in the space of the category *Acquisitions*. The stars are the documents belonging to this category, while the circles are all the other documents

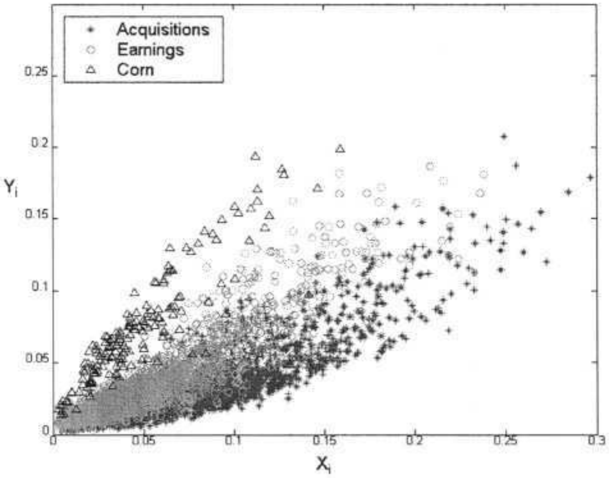


Fig. 7. Only the training documents belonging to the category *Acquisitions, Earnings, Corn* are plotted in the space of category *Acquisitions*

This peculiar behavior suggests that the projection preserves some kind of relation among categories that might be used in order to improve the performance of the categorisation system.

One final remark about the computational costs is needed. It is important to lay particular stress on the low complexity of projecting the documents into the bidimensional plane. From a computational point of view, calculating the Presence for all the categories is linearly proportional to the number of categories *C,* the number of documents *D* and the number of distinct terms *T,* $O(C \times D \times T \times h)$, assuming to have a structure like a HashMap with a constant access h. Computing the Expressiveness for all the categories is a quadratic problem with respect to the number of categories and linear with respect to the terms, $O(C^2 \times T \times h)$ (see [17] for the details on the computational costs).

This is a very efficient algorithm, which is almost comparable to a feature selection step in a common categorisation system.

5 Future Work

It is our intent on completing some compelling issues in the future:

- first, we shall complete the experimentation augmenting the numbers of categories Reuters-21578, and testing on different benchmarks like the new RCV1 Reuters, or the 20NewsGroups;
- secondly, it is important for us to understand how to use the information given by overlapping categories, eventually iterating the process of computing the energy on a particular subset of the categories;

- then, we might explore the possibility of extending the notion of Presence including the term frequencies within a document and/or weighting differently the Presence and Expressiveness, for example $P^{\alpha} \cdot E^{\beta}$ where α and β are two parameters;
- finally, we shall explore the possibility of using Presence and Expressiveness as feature selection functions.

6 Conclusions

This paper provides a technique for projecting documents into a Cartesian plane. This technique proposes a new concept of term weighting, localizing the word in the context of the category of interest and in the context of the other categories, replacing the *tf*×*idf* weighting scheme. Through it, the concept of both local and global energy of a category and the energy of a document is presented.

A heuristic learning method that uses the information of the bidimensional information has been compared with other state-of-the-art methods. The results were satisfactory showing that this compact graphical representation does summarize well the content of the high-dimensional space. Moreover, a more accurate investigation shows that the distribution of documents gives some clues about some relations among categories that might be exploited in order to improve the performance of the categorisation system.

Acknowledgment. The author would like to give the most sincere thanks to Massimo Melucci and Luca Pretto for the fruitful discussions and suggestions. Thanks to Claudio Biancalana for running the support vector machines for the experimental results. Also thanks to the anonymous reviewers for their helpful comments.

References

1. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval **1** (1999) 69–90
2. Yang, Y., Liu, X.: A re-examination of text categorization methods. In Gey, F., Hearst, M., Tong, R., eds.: Proceedings of the Twenty-Second Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR–99), Berkeley, California, US, ACM Press, New York, US (1999) 42–49
3. Eyheramendy, S., Lewis, D.D., Madigan, D.: On the naive bayes model for text categorization. In: Proceedings of the Ninth International Workshop Artificial Intelligence and Statistics (AISTATS–03), Key West, Florida, US (2003)
4. Lewis, D.D.: Naive (bayes) at forty: The independence assumption in information retrieval. In Nédellec, C., Rouveirol, C., eds.: Proceedings of the Tenth European Conference on Machine Learning (ECML–98), Chemnitz, DE, Springer Verlag, Heidelberg, DE (1998) 4–15
5. Zang, T., Oles, F.J.: Text categorization based on regularized linear classification methods. Information Retrieval **4** (2001) 5–31

6. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In Nédellec, C., Rouveirol, C., eds.: Proceedings of the Tenth European Conference on Machine Learning (ECML–98), Chemnitz, DE, Springer Verlag, Heidelberg, DE (1998) 137–142
7. Aas, K., Eikvil, L.: Text categorisation: A survey. Technical Report NR 941, Norwegian Computing Center, Oslo (1999)
8. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47
9. Galavotti, L., Sebastiani, F., Simi, M.: Experiments on the use of feature selection and negative evidence in automated text categorization. In Borbinha, J.L., Baker, T., eds.: Research and Advanced Technology for Digital Libraries, Fourth Eurpean Conference (ECDL–00). Volume 1923 of Lecture Notes in Computer Science., Lisbon, Portugal, Springer Verlag, Heidelberg, DE (2000) 59–68
10. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In Fisher, D.H., ed.: Proceedings of the Fourteenth International Conference on Machine Learning (ICML–97), Nashville, Tennessee, US, Morgan Kaufmann Publishers, San Francisco, US (1997) 412–420
11. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407
12. Berry, M.W., Dumais, S.T., O'Brien, G.W.: Using linear algebra for intelligent information retrieval. SIAM Review **37** (1995) 573–595
13. Kohonen, T.: Self-Organizing Maps. Springer Verlag, Berlin and Heidelberg, DE (1995)
14. Kruskal, J.B., Wish, M.: Multidimensional Scaling. Sage University Paper Series on Quantitative Applications in the Social Sciences. Sage Publications, London, UK (1978)
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management **24** (1988) 513–523
16. Debole, F., Sebastiani, F.: Supervised term weighting for automated text categorization. In: 18th ACM Symposium on Applied Computing, Melbourne, US, ACM Press, New York, US (2003) 784–788
17. Di Nunzio, G.M., Micarelli, A.: Does a new gaussian weighting approach perform well in text categorization? In Gottlob, G., Walsh, T., eds.: Proceedings of the Eighteenth International Joint Conference of Artificial Intelligence (IJCAI–03), Acapulco, Mexico, Morgan Kaufmann Publishers, San Francisco, US (2003) 581–586
18. Zobel, J., Moffat, A.: Exploring the similarity space. SIGIR Forum **32** (1998) 18–34
19. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer Series in Statistics. Springer Verlag, New York, US (1995)
20. van Rijsbergen, C.J.: Information Retrieval. Second edn. Dept. of Computer Science, University of Glasgow, Glasgow, UK (1979)
21. Eyheramendy, S., Gerkin, A., Ju, W.H., Lewis, D.D., Madigan, D.: Sparse bayesian classifiers for text categorization. Sumbitted to JICRD, available at www.stat.rutgers.edu/~madigan/PAPERS/jicrd-v13.pdf (2003)

Query Difficulty, Robustness, and Selective Application of Query Expansion

Giambattista Amati, Claudio Carpineto, and Giovanni Romano

Fondazione Ugo Bordoni
Rome Italy
{gba,carpinet,romano}@fub. it

Abstract. There is increasing interest in improving the robustness of IR systems, i.e. their effectiveness on difficult queries. A system is *robust* when it achieves both a high Mean Average Precision (MAP) value for the entire set of topics and a significant MAP value over its worst X topics (MAP(X)). It is a well known fact that Query Expansion (QE) increases global MAP but hurts the performance on the worst topics. A selective application of QE would thus be a natural answer to obtain a more robust retrieval system.

We define two information theoretic functions which are shown to be correlated respectively with the average precision and with the increase of average precision under the application of QE. The second measure is used to selectively apply QE. This method achieves a performance similar to that with unexpanded method on the worst topics, and better performance than full QE on the whole set of topics.

1 Introduction

Formulating a well-defined topic is a fundamental issue in Information Retrieval. Users in general are not aware of the intrinsic ambiguity conveyed by their queries, as well as they are not confident on whether submitting short or long queries to obtain the highest retrieval quality. It is a largely accepted evidence that, for example, pseudo-relevance feedback (also known as blind feedback or retrieval feedback) can be used to expand original queries with several additional terms with the aim of a finer formulation of the initial queries.

In many cases the QE process succeeds, but in some cases QE worsens the quality of the retrieval. Global performance values tell us the average behaviour of the system, but not if the system has a large variance in performance over all single topics. Retrieval can be excellent with some queries and very poor-performing with others. The introduction of the notion of robustness in retrieval is thus motivated by the necessity of improving the quality of the retrieval also on the most difficult queries.

Two new evaluation measures for robustness have been defined in the TREC environment, i.e. the number of topics with no relevant documents in the top retrieved 10 (denoted in this paper by NrTopicsWithNoRel) and MAP(X), which measures the area under the average precision over the *worst* X (=%25 of) topics. A topic is deemed worst with respect to the individual run being evaluated.

The use of full QE usually results in an improvement of global MAP over the unexpanded method. However, we notice that:

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 127–137, 2004.

- the number NrTopicsWithNoRel of topics with no relevant documents in the top retrieved 10 increases when QE is activated, and similarly
- MAP of the worst X topics diminishes when QE is adopted.

Briefly, QE always worsen the performance of the system on its poor-performing queries. The objective of our study focuses on defining a decision method for QE activation able to achieve or ameliorate as much as possible the global MAP value obtained by full QE, while keeping the other two new measures at the same value as the unexpanded method.

We address the following issues:

- defining an operational notion of a poor-performing query, that can be used to have a measure on the risk of performing QE on that query.
- defining a measure predicting on what queries there is a reasonable chance that QE fails or succeeds. This measure can be used to selectively activate QE.

The problem of predicting a poor-performing query is not new. It has been investigated under different names, such as query-difficulty, query-specificity, query-ambiguity or even as an inherent problem of QE. Indeed, the importance of the query-terms based on the quality of the first-pass ranking can be assessed. According to Kwok [8] the within-document term-frequency and the standard *Idf* can be combined to “peak up” those query-terms that hold a higher weight in the original query. Difficult queries are those which do not possess a variety of such important terms.

A different approach uses a similarity measure based on the cosine function to generate the query-space. This topology does not exhibit a significant regularity in the position of the difficult and of the easy queries. There is only some continuity in the position of the queries of the same type [10].

The closest work to our investigation is the clarity score based on the language model [6]. The clarity score needs to be computed at the indexing time since it contains a collection model component.

As far as we know there has not been any methodological or experimental work addressing the problem of the selective application of QE. This is a challenging task because it requires to formally relate the relevance to other notions like query-difficulty and query expansion.

We propose a method which achieves a performance similar to that of the unexpanded baseline on the worst topics, but better performance than full QE on the whole set of topics. Our work is thus a fist step towards the definition of a decision methodology for the selective use of the QE.

In our analysis we use the DFR *(Divergence From Randomness)* modular probabilistic framework [2,3,1] together with query expansion based on distribution analysis [4, 5,1] for retrieval. We use the data of the TREC 2003 robust track.

2 Retrieval Setting

Our objective is to develop a methodology for a stable and robust QE activation to improve the performance on both the worst topics and all topics. We consider the description-only queries of the TREC 2003 robust track data. Their average length after stemming and with the stop list is about 8 terms.

The retrieval framework is made up of two components: the DFR within-document term-weighting model and QE within-query term-weighting model. In the next two sections we briefly describe these two components. We use different DFR models to test robustness with selective QE activation.

2.1 Term-Weighting Models

The DFR within-document term-weighting models are:

I(n)OL2, I($\mathrm{n_e}$)OL2, I(n)B2, I($\mathrm{n_e}$)B2, I($\mathrm{n_e}$)OB2. They are obtained from the generating formula:

$$\mathrm{Info_{DFR}} = -\log_2 \mathrm{Prob}(\mathrm{term\_freq}|\mathrm{doc\_freq}, \mathrm{Freq}(\mathrm{term}|\mathrm{Collection})) \qquad (1)$$

where Prob is the probability of obtaining a given within-document term-frequency randomly. Formula 1 is not used directly, but it is normalized by considering the probability of the observed term-frequency only in the set of documents containing the term. The final weighting formulas are:

$$\mathrm{I(n)OL2}: \quad \frac{tfn}{tfn+1}\log_2\left(\frac{N-\mathrm{doc\_freq}+1}{\mathrm{doc\_freq}+0.5}\right) \qquad (2)$$

$$\mathrm{I(n_e)OL2}: \quad \frac{tfn}{tfn+1}\log_2\left(\frac{N-n_e+1}{n_e+0.5}\right) \qquad (3)$$

$$\mathrm{I(n)B2}: \quad \frac{\mathrm{Freq}(\mathrm{term}|\mathrm{Collection})+1}{\mathrm{doc\_freq}\cdot(tfn+1)}\left(tfn\cdot\log_2\frac{N+1}{\mathrm{doc\_freq}+0.5}\right) \qquad (4)$$

$$\mathrm{I(n_e)B2}: \quad \frac{\mathrm{Freq}(\mathrm{term}|\mathrm{Collection})+1}{\mathrm{doc\_freq}\cdot(tfn+1)}\left(tfn\cdot\log_2\frac{N+1}{n_e+0.5}\right) \qquad (5)$$

$$\mathrm{I(n_e)OB2}: \quad \frac{\mathrm{Freq}(\mathrm{term}|\mathrm{Collection})+1}{\mathrm{doc\_freq}\cdot(tfn+1)}\left(tfn\cdot\log_2\frac{N-n_e+1}{n_e+0.5}\right) \qquad (6)$$

where

$tfn = \mathrm{term\_freq}\cdot\log_2\left(1+c\cdot\frac{\mathrm{average\_document\_length}}{\mathrm{document\_length}}\right)$,

N is the size of the collection,

$n_e = N\cdot\left(1-\left(\frac{1}{N}\right)^{\mathrm{Freq(term|Collection)}}\right)$,

$\mathrm{Freq}(\mathrm{term}|\mathrm{Collection})$ is the within-collection term-frequency,

term_freq is the within-document term-frequency,

doc_freq is the document-frequency of the term,

the parameter c is set to 3.

2.2 Query Expansion

The QE method is the same as used an TREC-10 with very good results[2] except for the parameter tuning and some additional expansion weight models.

The weight of a term of the expanded query q^* of the original query q is obtained as follows:

$$\text{weight}(\text{term} \in q^*) = tfq_n + \beta \cdot \frac{\text{Info}_{\text{DFR}}}{\text{MaxInfo}}$$

where

tfq_n is the normalized term-frequency within the original query q, i.e. $\frac{tfq}{max_{t \in q} tfq}$

$\text{MaxInfo} = \arg_{t \in q^*} \max \text{Info}_{\text{DFR}}$

Info_{DFR} is a term-frequency in the expanded query induced by using a DFR model, that is:

$$\text{Info}_{\text{DFR}} = -\log_2 \text{Prob}(\text{Freq}(\text{term}|\text{TopDocuments})|\text{Freq}(\text{term}|\text{Collection})) \qquad (7)$$

Formula 7 uses the same probabilistic model Prob of Formula 1, but the observed frequencies are different. The term-weighting models compute the probability of obtaining a given within-document term-frequency, whereas the within-query term-weighting computes the probability of obtaining a given term-frequency within the topmost retrieved documents.

For the implementation of Info_{DFR} we here use the normalized Kullback-Leibler measure (KL) [4,2]

$$\text{Info}_{\text{KL}}(\text{t}) = \frac{\text{Freq}(\text{t}|\text{TopDocs})}{\text{TotFreq}(\text{TopDocs})} \cdot \log_2 \frac{\text{Freq}(\text{t}|\text{TopDocs}) \cdot \text{TotFreq}(\text{C})}{\text{TotFreq}(\text{TopDocs}) \cdot \text{Freq}(\text{t}|\text{C})} \qquad (8)$$

where C indicates the whole collection and TopDocs denotes the pseudo-relevant set, while the Bose-Einstein statistics (Bo2) is:

$$\begin{aligned}\text{Info}_{\text{Bo2}}(\text{t}) &= -\log_2\left(\tfrac{1}{1+\lambda}\right) - \text{Freq}(\text{t}|\text{TopDocuments}) \cdot \log_2\left(\tfrac{\lambda}{1+\lambda}\right) \quad [\text{Bo2}] \\ \lambda &= \text{TotFreq}(\text{TopDocuments}) \cdot \frac{\text{Freq}(\text{t}|\text{Collection})}{\text{TotFreq}(\text{Collection})} \qquad (9)\end{aligned}$$

A further condition imposed for the selection of the new query-terms is that they must appear in at least two retrieved documents. This condition is to avoid the noise that could be produced by those highly informative terms which appear only once in the set of the topmost retrieved documents. The QE parameters are set as follows:

$\beta = 0.4$

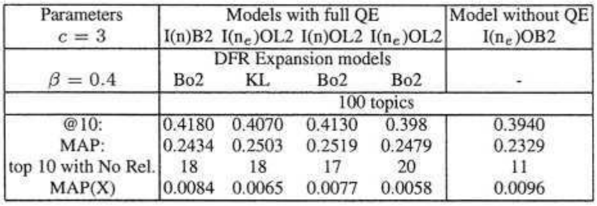
$|\text{TopDocuments}| = 10$

the number of terms of the expanded query is equal to 40.

Table 1 compares a baseline run with the full QE runs. We chose the model $\text{I}(\text{n}_\text{e})\text{OB2}$ defined in Formula 6 as baseline for the comparison, since it is the " best" performing model on the most difficult topics.

The unexpanded runs achieve the best MAP(X) and the lowest NrTopicsWithNoRel, and the runs with expanded queries achieve the highest values ofMAP and precision at 10.

Table 1. The number of selected documents on the first-pass retrieval is 10, the number of the extracted terms for query expansion is 40.



Parameters	Models with full QE				Model without QE
c = 3	I(n)B2	I(n_e)OL2	I(n)OL2	I(n_e)OL2	I(n_e)OB2
	DFR Expansion models				
$\beta = 0.4$	Bo2	KL	Bo2	Bo2	-
	100 topics				
@10:	0.4180	0.4070	0.4130	0.398	0.3940
MAP:	0.2434	0.2503	0.2519	0.2479	0.2329
top 10 with No Rel.	18	18	17	20	11
MAP(X)	0.0084	0.0065	0.0077	0.0058	0.0096

3 Selective Application of QE

In the following we study the problem of selectively applying QE to the set of topics.

We exploit the **Info**$_{\mathbf{DFR}}$ measures, as defined by Formula 7, and introduce a new measure InfoQ. We show that the sum of all **Info**$_{\mathbf{DFR}}$ over the terms of the query is related to the Average Precision (AP) and InfoQ is related to the AP increase after the QE activation. In other words, **Info**$_{\mathbf{DFR}}$ is an indicator of a possible low outcome of AP, attesting thus when a topic is possibly *difficult.* On the other hand, InfoQ is an indicator of the successful application of QE.

These findings can be a first step towards the definition of a more stable decision strategy for the selective use of the QE.

3.1 Test Data

The document collection used to test robustness is the set of documents on both TREC Disks 4 and 5 minus the Congressional Record on disk 4, containing 528,155 documents of Ê1.9 GB size. The set of test-topics contains 100 statements. Among these topics there are 50 topics that are known to be difficult for many systems. These 50 difficult topics were extracted from all 150 queries of previous TRECs using this same collection. We have indexed all fields of the documents and used Porter's stemming algorithm.

3.2 How QE Affects Robustness

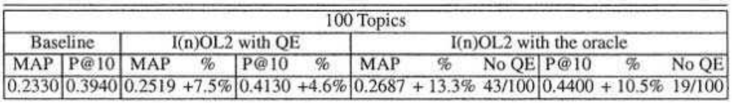
Consider as an example the performance of the model of Formula 2, I(n)OL2, as shown in Table 2.

With full QE, we achieve an increase of MAP equal to +7.5% with respect to the baseline run. If we had an oracle telling us when to apply QE query-by-query, the MAP increase would nearly double passing from +7.5% to +13.3%.

However, without the oracle a wrong decision of omitting the QE mechanism would seriously hurt the final MAP of the run. The average gain per query is ~0.063 and the gain is much greater than the average loss (~0.039). Moreover, the number of cases with a successful application of QE (57 out 100) is larger than the number of the failure cases. Both odds are thus in favour of the application of QE.

Comparing the figures of Table 2 with those relative to all the 150 queries of the past TREC data, we observe a detriment of the success rate. The success rate is around 65%

Table 2. Run I(n)OL2 with description-only topics. The columns with "No QE" contain the number of queries to which the QE was not applied.



100 Topics											
Baseline		I(n)OL2 with QE				I(n)OL2 with the oracle					
MAP	P@10	MAP	%	P@10	%	MAP	%	No QE	P@10	%	No QE
0.2330	0.3940	0.2519	+7.5%	0.4130	+4.6%	0.2687	+ 13.3%	43/100	0.4400	+ 10.5%	19/100

with all the 150 old queries of past TREC data. A detriment in precision at 10 is observed for only 15% of all the 150 old queries (against 19% of the TREC 2003 queries).

In addition, the increase of MAP with QE using all the old 150 queries is larger (~ +10%) than that obtained with this TREC data (~ +5%).

3.3 Selective Application of QE: Topic Difficulty

It is a well known evidence that the QE effectiveness is strictly related to the number of documents which are relevant for a given query in the set of the topmost documents in the ranking. If the early precision of the first-pass retrieval is high, then we have a good chance to extract good additional topic terms together with their relative query-weights. To start our investigation we first compute the correlation factor between

- the number *Rel* of relevant documents in the whole collection and the AP value over the 100 queries, and
- between *Rel* and the precision at 10(P@ 10).

The correlation value $-1 \leq \rho \leq 1$ indicates the degree of the linear dependence between the two pair of measurements. When $\rho = 0$ the correlation coefficient indicates that the two variables are independent. When instead there is a linear correlation, the correlation coefficient is either –1 or 1 [7]. A negative correlation factor indicates that the two variables are inversely related.

Surprisingly, these correlation factors come out to be both negative:
$\rho(Rel, \mathrm{AP}) = -0.36$ and $\rho(Rel, \mathrm{P@10}) = -0.14$.

Although in these two cases the absolute values of the correlation coefficient are not close to –1, even small values of the correlation factor are regarded very meaningful especially in large samples [9].

Therefore, these values of the correlation factors seem to demonstrate that the greater the number *Rel* of relevant documents, the less the precision (MAP and P@10). An approximation line of the scatter line of the AP values for different numbers of relevant documents is produced in Figure 1. The fact that the correlation factor with AP is larger than that with P@ 10 is due to the definition of AP. The AP measure combines recall and precision by using the number *Rel* of relevant documents.

This negative correlation might appear to be counter-intuitive, since among the easiest topics there are many which possess a small number of relevant documents, and, as opposite, many difficult topics have many relevant documents. On the other hand, a possible explanation of these negative correlation factors is that a small number of relevant documents for a topic witnesses the fact that the topic is "specific" or "non-general" with respect to the content of the collection. In such a situation, common-sense

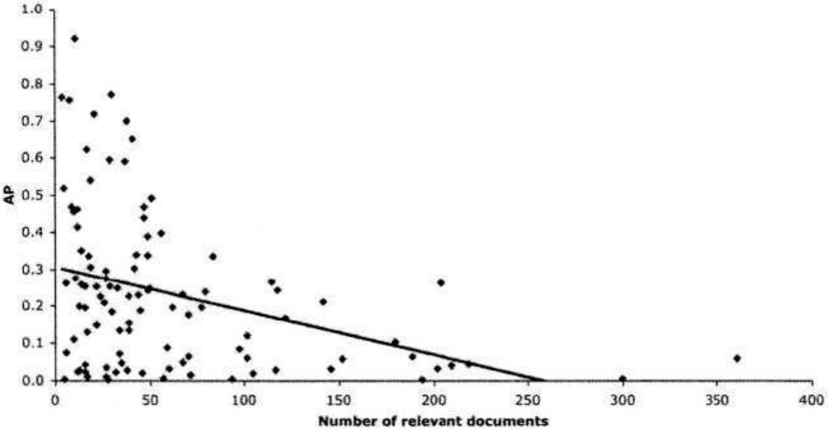


Fig. 1. The number of relevant documents is inversely related to AP of the unexpanded query ($\rho(Rel, \mathrm{AP}) = -0.36$). Queries with many relevant documents contribute little to MAP.

says that specific queries have few relevant documents, their query-terms have few occurrences in the collection, and they thus are the easiest ones.

However, a definition of the specificity based on the number of relevant documents for the query would depend on the evaluation; we rather prefer to have a different but operational definition of the query-specificity or query-difficulty.

The notion of query-difficulty is given by the notion of the amount of information $\mathrm{Info}_{\mathrm{DFR}}$ gained after a first-pass ranking. If there is a significant divergence in the query-term frequencies before and after the retrieval, then we make the hypothesis that this divergence is caused by a query which is easy-defined.

$$\mathrm{Info}_{\mathrm{DFR}} = \sum_{t \in Q} -\log_2 \mathrm{Prob}(\mathrm{Freq}(t|\mathrm{TopDocuments})|\mathrm{Freq}(t|\mathrm{Collection})) \qquad (10)$$

where $\mathrm{Prob}(t|\mathrm{Collection}$,TopDocuments) is a DFR basic model (based on the Binomial, the Bose-Einstein statistics or the Kullback-Leibler divergence measure, as defined in Formulas 8 and 9). We here use the probability of Bose-Einstein defined in Formula (9). We stress agin the fact that the same weighting formula is used by our expansion method, together with the Kullback-Leibler divergence $\mathrm{I(n_e)OL2}$ (see Table 1).

There are other information theoretic measures capturing the notion of term–specificity of the query.

The goodness of $\mathrm{Info}_{\mathrm{DFR}}$ is tested with the linear correlation factor with AP of the unexpanded queries. The motivation is that easy queries usually yield high AP values. To compute the difficulty score of the query we first produced a first-pass ranking as it is done in QE. We took the set TopDocuments of the first 10 retrieved documents and we computed a score for each term occurring in the query. We considered the query-terms which appear at least twice in these pseudo-relevant documents. This score reflects the amount of information carried by the query-term within these pseudo-relevant

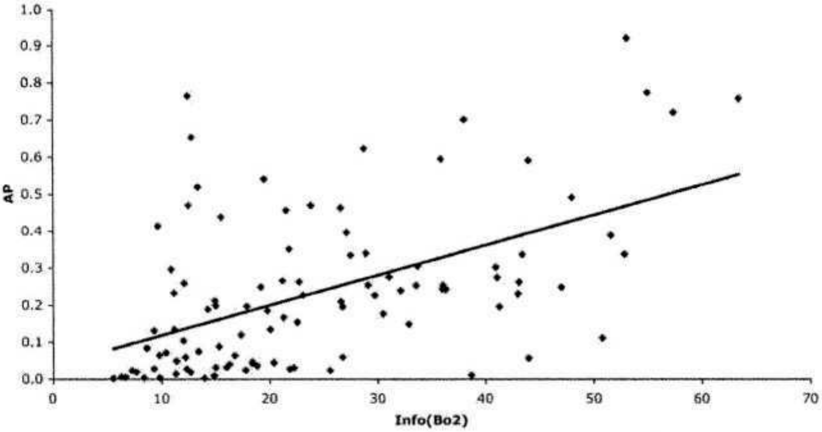


Fig. 2. The information content Info$_{\mathrm{Bo2}}$ of the query within the topmost retrieved documents is linearly correlated to the AP of the unexpanded queries ($\rho(\text{Info}_{\mathrm{Bo2}}, \text{AP}) = 0.52$). Specific queries have a large value of Info$_{\mathrm{DFR}}$.

documents. As shown in Figure 2, Info$_{\mathrm{DFR}}$ has a significant correlation with the AP of the unexpanded queries $\rho(\text{Info}_{\mathrm{Bo2}}, \text{AP}) = 0.52$. Similarly to the negative correlation between the number of relevant documents and the AP of the unexpanded queries, which is $\rho(Rel, \text{AP}) = -0.36$, the correlation factor between the score InfoQ and Rel is negative ($\rho(Rel, \text{Info}_{\mathrm{Bo2}}) = -0.23$). Again, this may be explained by the fact that specific queries possess fewer relevant documents.

Unfortunately, we did not find a significant correlation between Info$_{\mathrm{DFR}}$ and QE; that is, Info$_{\mathrm{DFR}}$ is not able to predict a successful application of QE in a second-pass ranking. These results show that the performance of query expansion is not directly related to query difficult, consistent with the observation [5] that although the retrieval effectiveness of QE in general increases as the query difficult decreases, very easy queries hurt performance.

3.4 Predicting the Successful Application of QE

Since Info$_{\mathrm{DFR}}$ cannot be used as a good indicator for the performance of the QE, we explore alternative information-theoretic functions. The function

$$\text{InfoPriorQ} = \sum_{\text{term} \in Q} -\log_2 \frac{\text{Freq}(\text{term}|\text{Collection})}{\text{TotFreq}(\text{Collection})}$$

is shown to have a moderately weak negative correlation with QE:
$\rho(\text{QE, InfoPriorQ}) = -0.27$.

InfoPriorQ is linearly related to the length of the query with correlation factor $\rho(\text{QueryLength, InfoPriorQ}) = 0.90$, so that InfoPriorQ does not differ to much from the query length. In other words, the query length is an alternative good indicator for

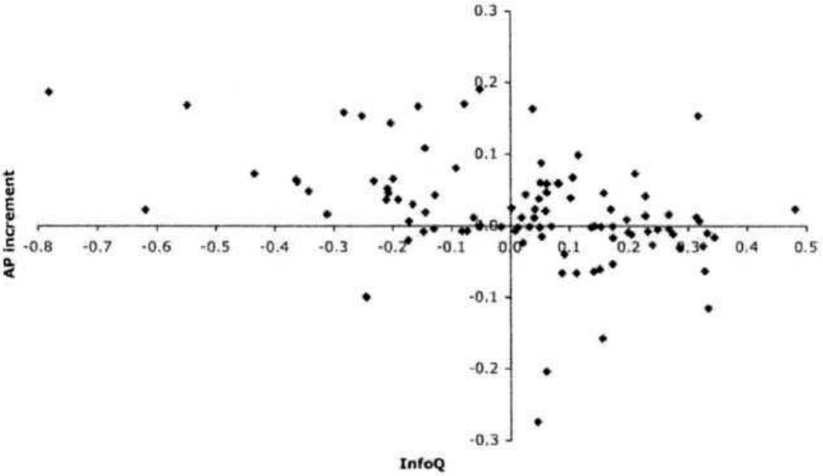


Fig. 3. The information content InfoQ of the query based on the combination of the priors and $\mathbf{Info}_{\mathbf{DFR}}$ within the topmost retrieved documents is negatively correlated to the AP increase with the $\mathbf{QE}$ ($\rho(\mathbf{QE\ increase\ rate},\ \mathbf{InfoQ}) = -0.33$). The first and the third quadrants contain the errors when the threshold is set to 0.

the successful application of the QE. Short queries need in general QE whilst very long queries do not need QE, but this simple fact does not solve the problem of moderately long queries for which QE may or may not succeed.

Let

$$M_Q = max\left(\frac{\text{InfoPriorQ} - \mu_{\text{InfoPriorQ}}}{\sigma_{\text{InfoPriorQ}}}, \max_{M \in \text{DFR}} \arg \frac{\text{Info}_{\text{DFR}} - \mu_{\text{Info}_{\text{DFR}}}}{\sigma_{\text{Info}_{\text{DFR}}}}\right)$$

The function:

$$\text{InfoQ} = \frac{1}{\text{QueryLength}}\left(\frac{\text{InfoPriorQ} - \mu_{\text{InfoPriorQ}}}{\sigma_{\text{InfoPriorQ}}} + M_Q\right) \tag{11}$$

where the μ_Xs and the σ_Xs stand for the mean and the standard deviation of the X values, combines InfoPriorQ and $\mathbf{Info}_{\mathbf{DFR}}$. $\mathbf{Info}_{\mathbf{DFR}}$ query rankings may not agree using different DFR models. Because the correlation factor is negative, and since we trigger the QE when InfoQ is below a given threshold, a cautious way to smooth different $\mathbf{Info}_{\mathbf{DFR}}$ values is to compare the threshold to the maximum value of all these DFR models, InfoPriorQ included.

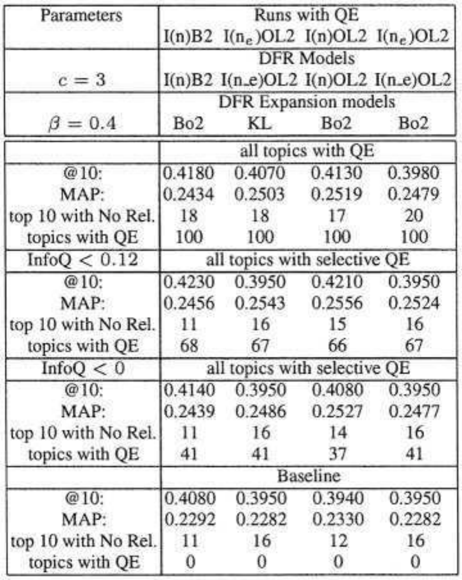
InfoQ has a higher correlation with $\mathbf{QE}$ ($\rho(\mathbf{QE},\ \mathbf{InfoQ}) = -0.33$) than InfoPriorQ (see Figure 3), and a smaller correlation factor with the query length[1] ($\rho(\mathbf{QE},\ \mathbf{InfoQ}) = 0.62$).

[1] Using $\log_2(\mathbf{QueryLength})$ instead of QueryLength the score of Formula 11 is more correlated to the query length with $\rho(\mathbf{QueryLength},\ \mathbf{InfoQ}) = 0.74$ and $\rho(\mathbf{QE},\ \mathbf{InfoQ}) = -0.34$.

Table 3. The set of queries with the highest InfoQ. The QE is not applied to such queries.

QE success	InfoQ	Query Length	Topic
y	0.482	7	604
n	0.345	8	631
n	0.335	17	320
n	0.333	13	638
n	0.329	9	621
n	0.327	14	619

Table 4. The selective application of QE.



Parameters	Runs with QE			
	I(n)B2	I(n_e)OL2	I(n)OL2	I(n_e)OL2
	DFR Models			
$c = 3$	I(n)B2	I(n_e)OL2	I(n)OL2	I(n_e)OL2
	DFR Expansion models			
$\beta = 0.4$	Bo2	KL	Bo2	Bo2
	all topics with QE			
@10:	0.4180	0.4070	0.4130	0.3980
MAP:	0.2434	0.2503	0.2519	0.2479
top 10 with No Rel.	18	18	17	20
topics with QE	100	100	100	100
InfoQ < 0.12	all topics with selective QE			
@10:	0.4230	0.3950	0.4210	0.3950
MAP:	0.2456	0.2543	0.2556	0.2524
top 10 with No Rel.	11	16	15	16
topics with QE	68	67	66	67
InfoQ < 0	all topics with selective QE			
@10:	0.4140	0.3950	0.4080	0.3950
MAP:	0.2439	0.2486	0.2527	0.2477
top 10 with No Rel.	11	16	14	16
topics with QE	41	41	37	41
	Baseline			
@10:	0.4080	0.3950	0.3940	0.3950
MAP:	0.2292	0.2282	0.2330	0.2282
top 10 with No Rel.	11	16	12	16
topics with QE	0	0	0	0

4 Discussion of Results

In Table 4 we summarize the results on the selective application of QE. The MAP(X) values are not reported since the new values are similar to those in the full QE models; thus we focus on the other measures. We compare the performance of models with full QE with the performance of the models with selective QE under the same setting.

The first remark is that the decision rule for QE activation is quite robust. The MAP of models with selective QE is greater than the MAP of the full QE models for a large range of values of the threshold parameter (> = 0). In fact, InfoQ provides with a high degree of confidence the cases in which QE should be absolutely activated, which are the cases when InfoQ assumes very small negative values, as it can be seen in Figure 3. This explains why the new value of MAP keeps constantly larger than the MAP obtained with all queries expanded. This decision method is thus safe.The behavior of Precision at 10 is more variable, depending on the choice of the threshold.

The second observation is that selective QE positively affects the NrTopicsWithNoRel measure. The models with selective QE have almost the same NrTopicsWithNoRel performance as the unexpanded runs, and this is one of the main objectives of our investigation.

5 Conclusions

We have defined two information theoretic functions used to predict the query-difficulty and to selectively apply QE. Our objective was to avoid the application of QE on the set of worst (difficult) topics. Indeed, QE application predictor achieves a performance similar to that of the unexpanded method on the worst topics, and better performance than full QE on the whole set of topics. Our work is thus a promising step towards a decision methodology for the selective use of the QE.

Acknowledgments. The experiments were conducted using the first version of the Terrier's Information Retrieval platform (http://www.dcs.gla.ac.uk/ir/terrier) initially developed by Gianni Amati during his PhD at Glasgow University. Terrier is a modular Information Retrieval framework that provides indexing and retrieval functionalities.

References

1. Giambattista Amati. *Probability Modelsfor Information Retrieval based on Divergencefrom Randomness.* PhD thesis, Glasgow University, June 2003.
2. Gianni Amati, Claudio Carpineto, and Giovanni Romano. FUB at TREC 10 web track: a probabilistic framework for topic relevance term weighting. In E.M. Voorhees and D.K. Harman, editors, *In Proceedings of the 10th Text Retrieval Conference TREC 2001,* pages 182–191, Gaithersburg, MD, 2002. NIST Special Pubblication 500-250.
3. Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS),* 20(4):357–389, 2002.
4. C. Carpineto, R. De Mori, G. Romano, and B. Bigi. An information theoretic approach to automatic query expansion. *ACM Transactions on Information Systems,* 19(1):1 –27, 2001.
5. C. Carpineto, G. Romano, and V. Giannini. Improving retrieval feedback with multiple term-ranking function combination. *ACM Transactions on Information Systems,* 20(3):259–290, 2002.
6. Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval,* pages 299–306. ACM Press, 2002.
7. Morris H. DeGroot. *Probability and Statistics.* Addison-Wesley, 2nd edition, 1989.
8. K. L. Kwok. A new method of weighting query terms for ad-hoc retrieval. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval,* pages 187–195. ACM Press, 1996.
9. Robert G.D. Steel, Jamies H. Torrie, and David A. Dickey. *Principles and Procedures of Statistics. A Biometrical Approach.* MacGraw–Hill, 3rd edition, 1997.
10. Terry Sullivan. Locating question difficulty through explorations in question space. In *Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries,* pages 251–252. ACM Press, 2001.

Combining CORI and the Decision-Theoretic Approach for Advanced Resource Selection

Henrik Nottelmann and Norbert Fuhr

Institute of Informatics and Interactive Systems, University of Duisburg-Essen,
47048 Duisburg, Germany,
{nottelmann,fuhr}@uni-duisburg.de

Abstract. In this paper we combine two existing resource selection approaches, CORI and the decision-theoretic framework (DTF). The state-of-the-art system CORI belongs to the large group of heuristic resource ranking methods which select a fixed number of libraries with respect to their similarity to the query. In contrast, DTF computes an optimum resource selection with respect to overall costs (from different sources, e.g. retrieval quality, time, money). In this paper, we improve CORI by integrating it with DTF: The number of relevant documents is approximated by applying a linear or a logistic function on the CORI library scores. Based on this value, one of the existing DTF variants (employing a recall-precision function) estimates the number of relevant documents in the result set. Our evaluation shows that precision in the top ranks of this technique is higher than for the existing resource selection methods for long queries and lower for short queries; on average the combined approach outperforms CORI and the other DTF variants.

1 Introduction

Today, there are thousands of digital libraries (DLs) in the world, most of them accessible through the WWW. For an information need, a decision must be made which libraries should be searched. This problem is called “library selection”, “collection selection”, “database selection” or “resource selection”. We use the latter term throughout this paper.

Recently several automatic selection methods have been proposed (see Sect. 2). In general they compute a ranking of libraries (based on similarities between the library and the query), and retrieve a constant number of documents from the top-ranked libraries. CORI [3] is one of the best performing resource ranking approaches.

In contrast to these heuristic methods, the decision-theoretic framework (DTF [6, 9] has a better theoretic foundation: The task is to find the selection with minimum costs (which depend on different criteria like retrieval quality, time or money). Thus, the system computes a clear cutoff for the number of libraries queried, and the number of documents which should be retrieved from each of these libraries. A user can choose different selection policies by specifying the importance of the different cost sources.

For DTF, different methods for estimating retrieval quality (i.e., the number of relevant documents in the result set) have been proposed. Here, we concentrate on DTF-rp. This method estimates the number of relevant documents in the complete DL, and uses

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 138–153, 2004.

a recall-precision function for computing the number of relevant documents in the result set. The quality of this resource selection variant is about the same as for CORI [9].

In this paper, we combine the advantages of both models: We use CORI for computing library scores, and map them with a linear or a logistic function onto the number of relevant documents in the complete DL. Then, this estimation is used in DTF-rp for estimating the number of relevant documents in the result set, and an optimum selection is computed by DTF. We investigate different approximations for the recall-precision function for both resulting "DTF-cori" variants as well as for DTF-rp.

One major advantage of our approach is that we extend the range of applications of CORI, as now other cost sources like time or money can be incorporated very easily.

The rest of this paper is organised as follows. First, we give an overview of related work, i.e. some other resource selection algorithms. Then, we describe CORI (Sec. 3) and the decision-theoretic framework (DTF, Sec. 4). In Sec. 5, we introduce our new, combined approach for resource selection with CORI and DTF. We compare this new approach with CORI and the two best performing DTF variants in Sec. 6. The last section contains some concluding remarks and an outlook to future work.

2 Related Work

Most of the resource selection algorithms follow the resource ranking paradigm: First, they compute a score for every library. Then, the top-ranked documents of the top-ranked libraries are retrieved and merged in a data fusion step. CORI (see Sec. 3) belongs to the resource ranking algorithms, whereas the decision-theoretic framework (DTF, see Sec. 4) does not rank DLs but explicitly computes for each library the number of documents which have to be returned.

The GlOSS system [7] is based on the vector space model and – thus – does not refer to the concept of relevance. For each library, a goodness measure is computed which is the sum of all scores (e.g. SMART scores) of all documents in this library w. r. t. the current query. Libraries are ranked according to the goodness values.

Other recent resource selection approaches are language models [14] (slightly better than CORI) and the cue validity variance model (CVV) [4] (slightly worse than CORI).

Query-based sampling is a technique for deriving statistical resource descriptions (e.g. average indexing weights, document frequencies) automatically in non-co-operating environments [1]. "Random" subsequent queries are submitted to the library, and the retrieved documents are collected. With reasonably low costs (number of queries), an accurate resource description can be constructed from samples of, e.g., 300 documents. Very recently, the problem of estimating the number of documents in a library (which in particular is important for DTF) has been investigated. Starting from query-based sampling, a sample-resample algorithm has been proposed in [13].

3 CORI

CORI is the state-of-the-art resource selection system [3,5]. CORI uses the inference network based system INQUERY for computing DL scores, and ranks the collections w. r. t. these scores.

INQUERY is an IR system, and as such, it ranks documents. In CORI, all documents in one collection are concatenated to form "meta-documents". As a consequence, the document nodes in the inference network are replaced by meta-documents, and the net has a moderate size. Therefore, resource selection with CORI is fast; for *N* collections, resource selection is equivalent to an IR run on *N* meta-documents. The frequency values are higher, but that does not affect computational complexity. A second advantage is that the same infrastructure can be used for both resource selection and document retrieval; there is no need for designing new index structures or algorithms.

Instead of the common *tf* · *idf* weighting scheme, *df* · *icf* is used, replacing the term frequency of a term by its document frequency *df*, the document frequency by the collection frequency *cf* (the number of libraries containing the term), and the document length by the collection length *cl* (the number of terms in the DL). Thus, the belief in a DL due to observing query term t (the "indexing weight" of term t in the "meta-document" DL) is determined by:

$$T := \frac{df}{df + 50 + 150 \cdot \frac{cl}{avgcl}} \tag{1}$$

$$I := \frac{\log(\frac{N+0.5}{cf})}{\log(N+1)} \tag{2}$$

$$Pr(t|DL) := 0.4 + 0.6 \cdot T \cdot I. \tag{3}$$

with *N* being the number of libraries which have to be ranked.

This indexing weighting scheme is quite similar to DTF's one (see next section), but applied to libraries instead of documents. As a consequence, in CORI the resource selection task is reduced to a document retrieval task on "meta-documents". The score of a DL depends on the query structure, but typically (and in this paper) it is the average of the beliefs $Pr(t|DL)$ for the query terms (i.e., a linear retrieval function is used with weight $1/ql$ for each query term).

CORI then selects the top-ranked libraries (the number of selected libraries is fixed before; typically, 10 DLs are chosen) and retrieves an equal number of documents from each selected DL.

CORI also covers the data fusion problem, where the library score is used to normalise the document score.

First the DL scores $C := Pr(q|DL)$ are normalised to [0, 1]:

$$C' := \frac{C - C_{min}}{C_{max} - C_{min}}, \tag{4}$$

where C_{min} and C_{max} are the minimum and maximum DL scores for that query.

Then, the document score $D := Pr(q|d)$ is normalised to D'' by

$$D' := \frac{D - D_{min}}{D_{max} - D_{min}}, \tag{5}$$

$$D'' := \frac{1.0 \cdot D' + 0.4 \cdot C' \cdot D'}{1.4}. \tag{6}$$

Finally, the retrieved documents are re-ranked according to the normalised scores D''.

4 Decision-Theoretic Framework

This section briefly describes the decision-theoretic framework (DTF) for resource selection [6,9].

4.1 Cost-Based Resource Selection

The basic assumption is that we can assign specific retrieval costs $C_i(s_i, q)$ to each digital library DL_i when s_i documents are retrieved for query q. The term "costs" is used in a broad way and also includes—besides money—cost factors like time and quality.

The user specifies (together with her query) the total number n of documents which should be retrieved. The overall number of all collections is denoted by m. The task then is to compute an optimum solution, i.e. a vector $\boldsymbol{s} = (s_1, s_2, \dots, s_m)^T$ which minimises the overall costs:

$$M(n,q) := \min_{|\boldsymbol{s}|=n} \sum_{i=1}^{m} C_i(s_i, q). \tag{7}$$

For $C_i(s_i, q)$, costs from different sources should be considered:

Effectiveness: Probably most important, a user is interested in getting many relevant documents. Thus we assign user-specific costs C^+ for viewing a relevant document and costs $C^- > C^+$ for viewing an irrelevant document. If $r_i(s_i, q)$ denotes the number of relevant documents in the result set when s_i documents are retrieved from library DL_i for query q, we obtain the cost function

$$C_i^{rel}(s_i, q) := r_i(s_i, q) \cdot C^+ + [s_i - r_i(s_i, q)] \cdot C^-. \tag{8}$$

Time: This includes computation time at the library site and communication time for delivering the result documents over the network. These costs can easily be approximated by measuring the response time for several queries. In most cases, a simple affine linear cost function is sufficient.

Money: Some DLs charge for their usage, and monetary costs often are very important for a user. These costs have to be specified manually. In most cases, the cost function is purely linear (per-document-charges).

All these costs are summed up to the overall cost function $C_i(s_i, q)$. With cost parameters C^+, C^-, C^t (time) and C^m (money), a user can specify her own selection policy (e.g. cheap and fast results). As the actual costs are unknown in advance, we switch to expected costs $EC_i(s_i, q)$ (for relevancy costs, using the expected number $E[r_i(s_i, q)]$ of relevant documents):

$$EM(n,q) := \min_{|\boldsymbol{s}|=n} \sum_{i=1}^{m} EC_i(s_i, q). \tag{9}$$

In formula 9, the expected costs $EC_i(s_i, q)$ are increasing with the number s_i of documents retrieved. Thus, the algorithm presented in [6] can be used for computing an optimum solution. Finally, all DLs with $s_i > 0$ are queried.

4.2 Retrieval Model

In this subsection we describe two methods for estimating retrieval quality, i.e. the expected number $E[r_i(s_i, q)]$ of relevant documents in the first s_i documents of a result set for all queries q. Both follow Rijsbergen's [15] paradigm of IR as uncertain inference, a generalisation of the logical view on databases. In uncertain inference, IR means estimating the probability $Pr(q \leftarrow d)$ that the document d logically implies the query q, where both d and q are logical formulae (set of terms with query term weights $Pr(q \leftarrow t)$ and indexing term weights $Pr(t \leftarrow d)$, respectively).

If we assume disjointness of query terms, we can apply the widely used linear retrieval function [17] for computing the probability of inference:

$$Pr(q \leftarrow d) := \sum_{t \in q} \underbrace{Pr(q \leftarrow t)}_{\text{query condition weight}} \cdot \underbrace{Pr(t \leftarrow d)}_{\text{indexing weight}} . \tag{10}$$

So far, this model does not cope with the concept of relevance. However, the decision-theoretic framework is based on estimates of the number $r(s, q)$ of relevant documents in the result set containing the first s documents. This number can be be computed using the probability $Pr(\text{rel}|q, d)$ that document d is relevant w. r. t. query q.

In [10], mapping functions have been proposed for transforming probabilities of inference into probabilities of relevance:

$$f : [0, 1] \mapsto [0, 1], \quad f_p(Pr(q \leftarrow d)) \approx Pr(\text{rel}|q, d). \tag{11}$$

Different functions can be considered as mapping functions; in previous work, linear and logistic functions have been investigated.

4.3 Estimating Retrieval Quality with Recall-Precision Function

Several methods for estimating retrieval quality have been developed within the decision-theoretic framework, where retrieval quality is measured as the expected number $E[r(s, q)]$ of relevant documents in the first s documents.

We only employ "DTF-rp" [6] in this work. This method first estimates the number of relevant documents in the complete DL. Then, a recall-precision function is used for computing the expected number of relevant documents in a result set.

DTF-rp is based on a linear mapping function [16]

$$f : [0, 1] \mapsto [0, 1], f(x) := c \cdot x \tag{12}$$

with constant $c := Pr(\text{rel}|q \leftarrow d)$.

We can compute the expected number $E(\text{rel}|q, DL)$ of relevant documents in DL as

$$E(\text{rel}|q, DL) = \sum_{d \in DL} Pr(\text{rel}|q, d) \tag{13}$$

$$= |DL| \cdot c \cdot \sum_{t \in q} Pr(q \leftarrow t) \cdot \mu_t \tag{14}$$

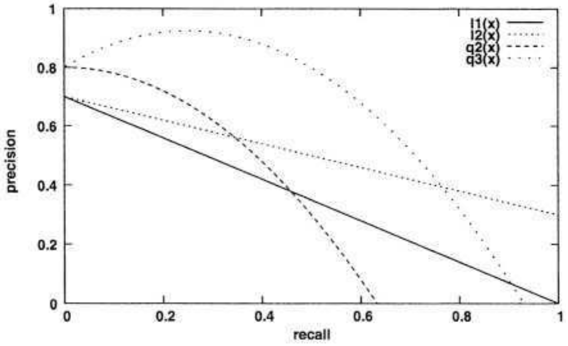


Fig. 1. Different recall/precision functions

with the average indexing weight of term t in DL

$$\mu_t := \frac{1}{|DL|} \sum_{d \in DL} Pr(t \leftarrow d). \tag{15}$$

We can assume different shapes of recall-precision functions, e.g. a linearly decreasing function with only one variable (called “11” in the remainder) or two degrees of freedom (“12”), or a quadratic function with two (“q2”) or three degrees of freedom (“q3”).

The shape of these recall-precision functions is depicted in Fig. 1. They are defined by:

$$P_{l1} : [0,1] \mapsto [0,1] \quad P_{l1}(R) := l_0 \cdot (1 - R) = l_0 - l_0 \cdot R, \tag{16}$$

$$P_{l2} : [0,1] \mapsto [0,1] \quad P_{l2}(R) := l_0 - l_1 \cdot R, \tag{17}$$

$$P_{q2} : [0,1] \mapsto [0,1] \quad P_{q2}(R) := q_0 - q_2 \cdot R^2, \tag{18}$$

$$P_{q3} : [0,1] \mapsto [0,1] \quad P_{q3}(R) := q_0 + q1 \cdot R - q_2 \cdot R^2. \tag{19}$$

Thus, P_{l1} is a special case of P_{l2} with $l_1 = l_0$, and P_{q2} is a special case of P_{q3} with $q1 = 0$.

Expected precision is defined as $EP := E[r(s,q)]/s$, expected recall as $ER := E[r(s,q)]/E(\text{rel}|q, DL)$.

So, when we assume a linear recall-precision function, we can estimate the number of relevant documents in a result set of s documents by

$$\frac{E[r(s,q)]}{s} = EP = P(ER) = l_0 - l_1 \cdot \frac{E[r(s,q)]}{E(\text{rel}|q, DL)}, \tag{20}$$

$$E[r(s,q)] := \frac{l_0 \cdot E(\text{rel}|q, DL) \cdot s}{E(\text{rel}|q, DL) + l_1 \cdot s}. \tag{21}$$

When we assume a quadratic recall-precision function, we have to solve the quadratic equation:

$$\frac{q_2}{E(\mathrm{rel}|q,DL)^2} \cdot E[r(s,q)]^2 + \left(\frac{1}{s} - \frac{q_1}{E(\mathrm{rel}|q,DL)^2}\right) E[r(s,q)] - q_0 = 0. \quad (22)$$

Thus, we can compute potential values for $E[r(s,q)]$ as:

$$p := \frac{E(\mathrm{rel}|q,DL)^2}{q_2 \cdot s} - \frac{q_1}{q_2} \cdot E(\mathrm{rel}|q,DL)^2, \quad (23)$$

$$q := \frac{q_0}{q_2} \cdot E(\mathrm{rel}|q,DL)^2, \quad (24)$$

$$E[r(s,q)] = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}. \quad (25)$$

5 Combining CORI and DTF

In this section we introduce a new method for estimating retrieval quality, called DTF-cori in the remainder of this paper. DTF-cori is similar to DTF-rp insofar as it also employs a recall-precision function, but here we estimate the number of relevant documents in the DL based on the CORI scores instead of formula 14.

CORI computes a ranking of libraries, based on DL scores $Pr(q|DL)$. Our basic assumption is that this score is related to the quality of the library. This is reasonable as the scores are used for ranking the libraries, and the system should favour high-quality DLs (w. r. t. the given query).

We are mainly interested in libraries containing many relevant documents in the top ranks. In Sect. 4.3 we presented DTF-rp, a technique for estimating the number of relevant documents in the top ranks based on the number $E(rel|q,DL)$ of relevant documents in the whole library. We can use this method by estimating $E(rel|q,DL)$ based on the DL score $Pr(q|DL)$ computed by CORI.

This is very similar to the case of single documents discussed in [10]. A retrieval engine computes a document "score" (called retrieval status value, RSV for short), and transforms it into the probability that this document is relevant. The relationship between score and probability of relevance is approximated by a mapping function.

In our setting, the retrieval engine is CORI, operating on "meta-documents" (the concatenation of all documents in a library). So, CORI computes a RSV $Pr(q|DL)$ for a meta-document *DL,* which has to be transformed into the probability $Pr(\mathrm{rel}|q,DL)$ that the DL is relevant (i.e., the average probability of relevance in that DL). Then, the expected number of relevant documents can easily be computed as:

$$E(rel|q,DL) = |DL| \cdot Pr(\mathrm{rel}|q,DL). \quad (26)$$

Similar to a mapping function, we introduce a "quality estimation function" which maps the CORI score $Pr(q|DL)$ of the DL onto the average probability of relevance in that DL, $Pr(\mathrm{rel}|q,DL)$:

$$f' : [0,1] \mapsto I\!R, \;\; f'(Pr(q|DL)) \approx Pr(\mathrm{rel}|q,DL). \quad (27)$$

In this paper, we start with a linear estimator (DTF-cori-lin). If we assume that the number of relevant documents in a library is proportional to the DL score, we arrive at a linear function. We can add one degree of freedom by using a constant part:

$$f'_{lin}(x) := c'_0 + c'_1 \cdot x. \tag{28}$$

We also investigate the use of logistic functions (DTF-cori-log). These functions perform well on the document level [10]:

$$f'_{log}(x) := \frac{\exp(b'_0 + b'_1 x)}{1 + \exp(b'_0 + b'_1 x)}. \tag{29}$$

As for the functions mapping document RSVs onto probabilities of relevance, these parameters are query-specific in general. However, as the number of relevant documents is unknown in advance, we only can learn DL-specific parameters.

6 Evaluation

This section describes our detailed evaluation of the decision-theoretic framework and its comparison with CORI.

6.1 Experimental Setup

As in previous resource selection evaluations, we used the TREC-123 test bed with the CMU 100 library split [1]. The libraries are of roughly the same size (about 33 megabytes), but vary in the number of documents they contain (from 752 to 33 723, the average is 10 782). The documents inside a library are from the same source and the same time-frame. All samples contain 300 documents.

We used the same document indexing terms and query terms (after stemming and stop word removal) for both CORI and the three DTF variants. The document index only contains the `<text>` sections of the documents. Queries are based on TREC topics 51–100 and 101–150 [8], respectively. We used three different sets of queries: short queries (`<title>` field, on average 3.3 terms, web search), mid-length queries (`<description>` field, on average 9.9 terms, advanced searchers) and long queries (all fields, on average 87.5 terms, common in TREC-based evaluations).

The standard weighting schemes for documents and queries are used for the CORI experiments. For the DTF experiments, a modified BM25 weighting scheme [12] is employed for documents:

$$P(t \leftarrow d) := \frac{tf(t,d)}{tf(t,d) + 0.5 + 1.5 \cdot \frac{dl(d)}{avgdl}} \cdot \frac{\log \frac{numdl}{df(t)}}{\log |DL|}. \tag{30}$$

Here, $tf(t,d)$ is the term frequency, $dl(d)$ denotes the document length (in terms), *avgdl* the average document length, *numdl* the sample or library size (number of documents), $|DL|$ the library size, and $df(t)$ the document frequency. We modified the standard BM25 formula by the normalisation component $1/\log|DL|$ to ensure that indexing weights are always in the closed interval [0,1] and can be regarded as a probability.

Normalised tf values are used as query term weights:

$$P(q \leftarrow t) := \frac{tf(t,q)}{ql(q)} \ . \tag{31}$$

Here, $tf(t,q)$ denotes the term frequency, and $ql(q) := \sum_{t \in q} tf(t,q)$ is the query length.

For DTF, we applied the same indexing and retrieval methods for the 100 libraries as we used for the resource selection index. We always requested 300 documents. For CORI, we employed the Lemur toolkit implementation[1] and selected 10 libraries (with 30 documents per selected DL) as in previous evaluations of Lemur[2]

The variants of the decision-theoretic framework (DTF-cori and DTF-rp) require a learning phase for bridging heterogeneous collections. The parameters are learned using a cross-evaluation strategy: Parameters are learned on TREC topics 51–100 and evaluated on topics 101–150, and vice versa. We used the Gnuplot[3] implementation of the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm [11] and the relevance judgements as probabilities of relevance for learning the parameters. As we don't have relevance judgements for all documents in practice, we only considered the 100 top-ranked documents.

6.2 Result Quality

The precision in the top ranks 5, 10, 15, 20 and 30 (averaged over all 100 topics) is depicted in Tab. 1–4, as well as the average precision.

The percentage values denote the difference to CORI; differences which are significant (assuming a t-Test with p=0.05) are marked with an asterisk.

When we assume a P_{l1} recall-precision function (linear function with one variable), then DTF-cori-lin outperforms DTF-cori-log, and both yield a better quality than CORI and DTF-rp in most cases. As already reported [9], DTF-rp also outperforms CORI in most cases.

Average precision for both DTF-cori variants is always higher than for CORI and DTF-rp. The difference is significant for DTF-cori-lin for all query types and for DTF-cori-log for all except short queries.

When we add a second degree of freedom, then DTF-cori-lin and DTF-cori-log still outperform DTF-rp (using the same recall-precision function), but quality significantly decreases compared to CORI. DTF-cori-lin is slightly better than DTF-cori-log. These results are surprising: In principle, adding one degree of freedom should increase the quality, as P_{l1} is a special case of P_{l2}. However, it seems that the parameters for P_{l2} fitted too much to the learning data ("overfitting").

When we employ a P_{q2} quadratic recall-precision function with 2 variables (i.e., it is monotonically decreasing), the DTF-cori quality is only slightly better compared to P_{l2}, but is still dramatically (and, in most cases, also significantly) worse than CORI.

[1] http://www-2.cs.cmu.edu/ lemur/

[2] The optimal constant number of selected libraries never has been evaluated for Lemur.

[3] http://www.ucc.ie/gnuplot/gnuplot.html

Table 1. Precision in top ranks and average precision, 11

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.4260 / +0.0%	0.3940 / -7.5%	0.3940 / -7.5%	0.4020 / -5.6%
10	0.3930 / +0.0%	0.3880 / -1.3%	0.3840 / -2.3%	0.3820 / -2.8%
15	0.3840 / +0.0%	0.3853 / +0.3%	0.3820 / -0.5%	0.3767 / -1.9%
20	0.3640 / +0.0%	0.3765 / +3.4%	0.3745 / +2.9%	0.3665 / +0.7%
30	0.3487 / +0.0%	0.3593 / +3.0%	0.3583 / +2.8%	0.3393 / -2.7%
Avg.	0.0517 / +0.0%	0.0730 / +41.2% *	0.0723 / +39.8%	0.0616 / +19.1%

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.3840 / +0.0%	0.4380 / +14.1%	0.4400 / +14.6%	0.4140 / +7.8%
10	0.3630 / +0.0%	0.4140 / +14.0%	0.4140 / +14.0%	0.3980 / +9.6%
15	0.3500 / +0.0%	0.4067 / +16.2%	0.4087 / +16.8%	0.3820 / +9.1%
20	0.3350 / +0.0%	0.3945 / +17.8%	0.3950 / +17.9%	0.3710 / +10.7%
30	0.3107 / +0.0%	0.3710 / +19.4%	0.3710 / +19.4%	0.3460 / +11.4%
Avg.	0.0437 / +0.0%	0.0716 / +63.8% *	0.0716 / +63.8% *	0.0509 / +16.5%

(b) Learned/evaluated on mid queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.5780 / +0.0%	0.5820 / +0.7%	0.5680 / -1.7%	0.5680 / -1.7%
10	0.5590 / +0.0%	0.5660 / +1.3%	0.5510 / -1.4%	0.5570 / -0.4%
15	0.5340 / +0.0%	0.5587 / +4.6%	0.5420 / +1.5%	0.5387 / +0.9%
20	0.5175 / +0.0%	0.5500 / +6.3%	0.5440 / +5.1%	0.5335 / +3.1%
30	0.5013 / +0.0%	0.5403 / +7.8%	0.5313 / +6.0%	0.5160 / +2.9%
Avg.	0.0883 / +0.0%	0.1371 / +55.3% *	0.1315 / +48.9% *	0.1029 / +16.5%

(c) Learned/evaluated on long queries

Finally, we evaluated all 3 DTF methods with a quadratic recall-precision function with 3 variables. For all DLs and query types, the system learned a quadratic function $q_0 + q_1 \cdot x - q_2 \cdot x^2$ with $q_2 < 0$. The results w. r. t. precision in the top ranks are heterogeneous: For short queries, both DTF-cori variants perform worse than CORI in the lower ranks and better in the higher ranks. For mid-length and long queries, DTF-cori outperforms CORI. Average precision of DTF-cori is better (except for short queries, significantly better) than for CORI. In all cases, DTF-cori performs slightly worse than in the case of a linear recall-precision function with 1 variable.

These results are also reflected in the corresponding recall-precision plots (which we leave out due to space restrictions).

Table 2. Precision in top ranks and average precision, 12

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.4260 / +0.0%	0.2820 / -33.8% *	0.2800 / -34.3% *	0.2400 / -43.7% *
10	0.3930 / +0.0%	0.2380 / -39.4% *	0.2350 / -40.2% *	0.1950 / -50.4% *
15	0.3840 / +0.0%	0.2027 / -47.2% *	0.1933 / -49.7% *	0.1627 / -57.6% *
20	0.3640 / +0.0%	0.1810 / -50.3% *	0.1730 / -52.5% *	0.1415 / -61.1% *
30	0.3487 / +0.0%	0.1477 / -57.6% *	0.1393 / -60.1% *	0.1187 / -66.0% *
Avg.	0.0517 / +0.0%	0.0085 / -83.6% *	0.0079 / -84.7% *	0.0097 / -81.2% *

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.3840 / +0.0%	0.2340 / -39.1% *	0.2060 / -46.4% *	0.1580 / -58.9% *
10	0.3630 / +0.0%	0.1980 / -45.5% *	0.1600 / -55.9% *	0.1190 / -67.2% *
15	0.3500 / +0.0%	0.1753 / -49.9% *	0.1380 / -60.6% *	0.0960 / -72.6% *
20	0.3350 / +0.0%	0.1530 / -54.3% *	0.1175 / -64.9% *	0.0775 / -76.9% *
30	0.3107 / +0.0%	0.1283 / -58.7% *	0.0920 / -70.4% *	0.0603 / -80.6% *
Avg.	0.0437 / +0.0%	0.0088 / -79.9% *	0.0042 / -90.4% *	0.0025 / -94.3% *

(b) Learned/evaluated on mid queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.5780 / +0.0%	0.4960 / -14.2%	0.5280 / -8.7%	0.2020 / -65.1% *
10	0.5590 / +0.0%	0.5000 / -10.6%	0.5140 / -8.1%	0.1550 / -72.3% *
15	0.5340 / +0.0%	0.4847 / -9.2%	0.4993 / -6.5%	0.1260 / -76.4% *
20	0.5175 / +0.0%	0.4715 / -8.9%	0.4735 / -8.5%	0.1070 / -79.3% *
30	0.5013 / +0.0%	0.4497 / -10.3%	0.4467 / -10.9%	0.0860 / -82.8% *
Avg.	0.0883 / +0.0%	0.0811 / -8.2%	0.0637 / -27.9% *	0.0043 / -95.1% *

(c) Learned/evaluated on long queries

6.3 Overall Retrieval Costs

Actual costs for retrieval (of 300 documents) and the number of selected DLs are shown in Tab. 5. Costs only refer to retrieval quality:

$$C_i(s_i, q) = s_i - r(s_i, q). \tag{32}$$

With a few exceptions, especially for the linear recall-precision function with two parameters, the costs for DTF-cori are lower than for CORI; in all cases, they are lower than DTF-rp.

On the other hand, DTF-cori selects a lot more DLs than CORI (always 10 DLs) and DTF-rp do; the number of selected DLs is maximal for the linear recall-precision function with one variable.

Table 3. Precision in top ranks and average precision, q2

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.4260 / +0.0%	0.2808 / -34.1% *	0.2404 / -43.6% *	0.3060 / -28.2% *
10	0.3930 / +0.0%	0.2838 / -27.8% *	0.2384 / -39.3% *	0.2930 / -25.4% *
15	0.3840 / +0.0%	0.2761 / -28.1% *	0.2290 / -40.4% *	0.2753 / -28.3% *
20	0.3640 / +0.0%	0.2667 / -26.7% *	0.2177 / -40.2% *	0.2560 / -29.7% *
30	0.3487 / +0.0%	0.2478 / -28.9% *	0.2007 / -42.4% *	0.2330 / -33.2% *
Avg.	0.0517 / +0.0%	0.0442 / -14.5%	0.0323 / -37.5% *	0.0357 / -30.9%

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.3840 / +0.0%	0.2820 / -26.6% *	0.2820 / -26.6% *	0.2300 / -40.1% *
10	0.3630 / +0.0%	0.2590 / -28.7% *	0.2620 / -27.8% *	0.2020 / -44.4% *
15	0.3500 / +0.0%	0.2447 / -30.1% *	0.2453 / -29.9% *	0.1840 / -47.4% *
20	0.3350 / +0.0%	0.2320 / -30.7% *	0.2310 / -31.0% *	0.1700 / -49.3% *
30	0.3107 / +0.0%	0.2170 / -30.2% *	0.2160 / -30.5% *	0.1657 / -46.7% *
Avg.	0.0437 / +0.0%	0.0341 / -22.0%	0.0336 / -23.1%	0.0161 / -63.2% *

(b) Learned/evaluated on mid queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.5780 / +0.0%	0.5020 / -13.1%	0.3580 / -38.1% *	0.3120 / -46.0% *
10	0.5590 / +0.0%	0.4750 / -15.0%	0.3280 / -41.3% *	0.3030 / -45.8% *
15	0.5340 / +0.0%	0.4660 / -12.7%	0.3220 / -39.7% *	0.2907 / -45.6% *
20	0.5175 / +0.0%	0.4525 / -12.6%	0.3140 / -39.3% *	0.2815 / -45.6% *
30	0.5013 / +0.0%	0.4420 / -11.8%	0.2980 / -40.6% *	0.2627 / -47.6% *
Avg.	0.0883 / +0.0%	0.1143 / +29.4%	0.0683 / -22.7%	0.0337 / -61.8% *

(c) Learned/evaluated on long queries

6.4 Approximation Quality

The mean square approximation error (linear recall-precision function with one variable) is depicted in Tab. 6. One can see that the linear estimator generates a significantly better approximation than DTF-cori-log and DTF-rp, where the latter one always heavily overestimates the number of relevant documents in the collection.

6.5 Evaluation Summary

From a theoretical point of view, integrating CORI into DTF has the advantage that other cost sources besides retrieval quality (e.g. time or money) can easily be incorporated. The evaluation results we reported in this section show that it also allows for better resource selections (on a theoretically founded basis) compared to the heuristic selection strategy of CORI ("select the 10 DLs with the highest scores and retrieve an equal amount of

Table 4. Precision in top ranks and average precision, q3

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.4260 / +0.0%	0.3860 / -9.4%	0.3820 / -10.3%	0.3660 / -14.1%
10	0.3930 / +0.0%	0.3760 / -4.3%	0.3740 / -4.8%	0.3350 / -14.8%
15	0.3840 / +0.0%	0.3753 / -2.3%	0.3727 / -2.9%	0.3160 / -17.7%
20	0.3640 / +0.0%	0.3695 / +1.5%	0.3655 / +0.4%	0.3050 / -16.2%
30	0.3487 / +0.0%	0.3507 / +0.6%	0.3497 / +0.3%	0.2807 / -19.5%
Avg.	0.0517 / +0.0%	0.0675 / +30.6%	0.0663 / +28.2%	0.0379 / -26.7%

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.3840 / +0.0%	0.4200 / +9.4%	0.4200 / +9.4%	0.3380 / -12.0%
10	0.3630 / +0.0%	0.3960 / +9.1%	0.3960 / +9.1%	0.3010 / -17.1%
15	0.3500 / +0.0%	0.3927 / +12.2%	0.3940 / +12.6%	0.2760 / -21.1%
20	0.3350 / +0.0%	0.3775 / +12.7%	0.3795 / +13.3%	0.2585 / -22.8%
30	0.3107 / +0.0%	0.3647 / +17.4%	0.3650 / +17.5%	0.2290 / -26.3% *
Avg.	0.0437 / +0.0%	0.0650 / +48.7% *	0.0648 / +48.3% *	0.0208 / -52.4% *

(b) Learned/evaluated on mid queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
5	0.5780 / +0.0%	0.5800 / +0.3%	0.5700 / -1.4%	0.4140 / -28.4% *
10	0.5590 / +0.0%	0.5660 / +1.3%	0.5610 / +0.4%	0.3940 / -29.5% *
15	0.5340 / +0.0%	0.5587 / +4.6%	0.5533 / +3.6%	0.3727 / -30.2% *
20	0.5175 / +0.0%	0.5485 / +6.0%	0.5480 / +5.9%	0.3435 / -33.6% *
30	0.5013 / +0.0%	0.5337 / +6.5%	0.5287 / +5.5%	0.2987 / -40.4% *
Avg.	0.0883 / +0.0%	0.1334 / +51.1% *	0.1315 / +48.9% *	0.0227 / -74.3% *

(c) Learned/evaluated on long queries

documents from each of these 10 DLs"). Precision both in the top ranks and on average is maximised by using DTF-cori-lin with a linear approximation (1 parameter) of the recall-precision function.

7 Conclusion and Outlook

In this paper, we combined the decision-theoretic framework [6,9] with CORI [3]. DTF has a better theoretic foundation (selection with minimum costs) than traditional resource ranking algorithms like CORI, considers additional cost sources like time and money, and computes the number of digital libraries to be queried as well as the number of documents which should be retrieved from each of these libraries. In contrast, heuristic methods like CORI compute a ranking of digital libraries, and additional heuristics are

Table 5. Actual costs and number of libraries selected

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
l1	245.7 / 10.0	237.3 / 69.2	238.3 / 70.1	239.0 / 40.5
l2	245.7 / 10.0	266.2 / 6.7	267.5 / 6.8	281.5 / 5.7
q2	245.7 / 10.0	248.7 / 30.3	241.9 / 25.8	256.0 / 15.6
q3	245.7 / 10.0	238.9 / 47.6	239.9 / 48.1	258.7 / 16.5

(a) Learned/evaluated on short queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
l1	256.8 / 10.0	241.7 / 67.6	241.8 / 68.6	254.6 / 28.1
l2	256.8 / 10.0	291.5 / 9.0	294.0 / 6.1	297.1 / 4.3
q2	256.8 / 10.0	270.3 / 22.6	270.6 / 22.8	282.8 / 11.9
q3	256.8 / 10.0	244.6 / 44.5	244.8 / 44.3	278.9 / 9.9

(b) Learned/evaluated on mid-length queries

	CORI	DTF-cori-lin	DTF-cori-log	DTF-rp
l1	229.0 / 10.0	205.5 / 66.0	211.4 / 69.0	226.1 / 29.7
l2	229.0 / 10.0	250.0 / 30.8	261.6 / 32.2	296.6 / 3.8
q2	229.0 / 10.0	226.7 / 39.7	252.6 / 30.3	272.9 / 9.8
q3	229.0 / 10.0	209.4 / 51.6	211.2 / 52.6	280.5 / 6.6

(c) Learned/evaluated on long queries

Table 6. Approximation error for number of relevant documents in DL

	DTF-cori-lin	DTF-cori-log	DTF-rp
short	110.49	123.81 / +12.1% *	140426.08 / >10^5% *
mid	96.33	122.62 / +27.3% *	527568.52 / >10^5% *
long	95.83	122.57 / +27.9% *	1585465.20 / >10^6% *

needed for determining the number of libraries and the number of documents to be retrieved. The retrieval quality of DTF is competitive with CORI.

Our new approach DTF-cori combines DTF and CORI. It first computes library scores with CORI which specify the similarity between the library and the query. This score is then mapped onto the expected number of relevant documents in the complete DL. We investigated the use of a linear and a logistic "estimation function" for this mapping. Then, the estimates of the number of relevant documents in the DL are used together with a recall-precision function (as for the DTF-rp variant) for approximating the number of relevant documents in the result set of given size. In this paper, we considered four different recall-precision functions: a linear one with one and two variables, and a quadratic one with two and three variables.

This new technique has two advantages: First, it extends the range of applications of CORI. Together with DTF, now other cost sources like time and money can also be incorporated in a natural way.

Second, the evaluation showed that we can increase precision both in the top ranks and on average when we integrate CORI into DTF. This indicates that DTF-cori can compute a better selection than CORI alone. The best results were obtained when a primitive linear function with only one variable and a linear estimation function is used. However, the differences in precision in the top ranks are not significant (in contrast to most differences in average precision).

When more degrees of freedom are allowed, we observe the effect of overfitting of the parameters to the learning data for a linear and the quadratic recall-precision function with two parameters each. The quadratic recall-precision function with three degrees of freedoms performs only slightly worse.

DTF-cori-lin approximates the number of relevant documents in a DL better than DTF-cori-log and DTF-rp whose estimates are much to high. This is only partially reflected by the retrieval quality, as the differences are not as high as suggested by the approximation errors.

In the future, we will have a look at better estimation functions. We are particularly interested in improving the retrieval quality for shorter queries, because this query type is commonly issued by users (e.g. on the web).

In addition, we will investigate the learning step of the estimation function parameters in more detail. In this paper, we learned parameters with 50 queries. The interesting question is how many documents per query and how many queries are really required for obtaining good parameters, and how the quality of the parameters is related to the size of the learning data. A major goal is to avoid overfitting.

Acknowledgements. This work is supported in part by the EU commission (grant IST-2000-26061, MIND), and in part by the DFG (grant BIB47 DOuv 02-01, PEPPER).

References

[1] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems,* 19(2):97–130, 2001.

[2] J. Callan, G. Cormack, C. Clarke, D. Hawking, and A. Smeaton, editors. *Proceedings of the 26st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* New York, 2003. ACM.

[3] J. Callan, Z. Lu, and W. Croft. Searching distributed collections with inference networks. In E. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* pages 21–29, New York, 1995. ACM. ISBN 0-89791-714-6.

[4] J. Callan, A. L. Powell, J. C. French, and M. Connell. The effects of query-based sampling on automatic database selection algorithms. *ACM Transactions on Information Systems (submitted for publication).*

[5] J. French, A. Powell, J. Callan, C. Viles, T. Emmitt, K. Prey, and Y. Mou. Comparing the performance of database selection algorithms. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval,* pages 238–245, New York, 1999. ACM.

[6] N. Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems,* 17(3):229–249, 1999.

[7] L. Gravano and H. Garcia-Molina. Generalizing GIOSS to vector-space databases and broker hierarchies. In U. Dayal, P. Gray, and S. Nishio, editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases,* pages 78–89, Los Altos, California, 1995. Morgan Kaufman.

[8] D. Harman, editor. *The Second Text REtrieval Conference (TREC-2),* Gaithersburg, Md. 20899,1994. National Institute of Standards and Technology.

[9] H. Nottelmann and N. Fuhr. Evaluating different methods of estimating retrieval quality for resource selection. In Callan et al. [2].

[10] H. Nottelmann and N. Fuhr. From uncertain inference to probability of relevance for advanced IR applications. In F. Sebastiani, editor, *25th European Conference on Information Retrieval Research (ECIR 2003),* pages 235–250, Heidelberg et al., 2003. Springer.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, editors. *Nested Relations and Complex Objects in Databases.* Cambridge University Press, 1992.

[12] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *Text REtrieval Conference,* pages 21–30,1992.

[13] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In Callan et al. [2].

[14] L. Si, R. Jin, J. Callan, and P. Ogilvie. Language model framework for resource selection and results merging. In D. Grossman, editor, *Proceedings of the 11th International Conference on Information and Knowledge Management,* New York, 2002. ACM.
http://www-2.cs.cmu.edu/ callan/Papers/cikm02-lsi.pdf.

[15] C. J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal,* 29(6):481–485, 1986.

[16] C. J. van Rijsbergen. Probabilistic retrieval revisited. *The Computer Journal,* 35(3):291–298, 1992.

[17] S. Wong and Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems,* 13(1):38–68, 1995.

Predictive Top-Down Knowledge Improves Neural Exploratory Bottom-Up Clustering

Chihli Hung[1,2], Stefan Wermter[1], and Peter Smith[1]

[1] Centre for Hybrid Intelligent Systems
School of Computing and Technology
University of Sunderland
Sunderland SR6 0DD, UK
http://www.his.sunderland.ac.uk
{chihli.hung,stefan.wermter,peter.smith}@sunderland.ac.uk
[2] De Lin Institute of Technology
Taipei, Taiwan
chihli@dlit.edu.tw

Abstract. In this paper, we explore the hypothesis that integrating symbolic top-down knowledge into text vector representations can improve neural exploratory bottom-up representations for text clustering. By extracting semantic rules from WordNet, terms with similar concepts are substituted with a more general term, the hypernym. This hypernym semantic relationship supplements the neural model in document clustering. The neural model is based on the extended significance vector representation approach into which predictive top-down knowledge is embedded. When we examine our hypothesis by six competitive neural models, the results are consistent and demonstrate that our robust hybrid neural approach is able to improve classification accuracy and reduce the average quantization error on 100,000 full-text articles.

1 Introduction

Document clustering is often performed under the assumption that predefined classification information is not available. Thus, the accuracy of clustering is mostly dependent on the definitions of cluster features and similarities since most clustering approaches organise documents into groups based on similarity measures. If the results of document clustering are compared with human classification knowledge, the accuracy depends on the difference between implicit factors of human classification assignment and explicit definitions of cluster features and similarities. However, pure unsupervised document clustering methods are sometimes unable to discern document classification knowledge hidden in the document corpus. One possible reason is that documents are classified not only on the basis of feature representation but also on the basis of human subjective concepts.

Clustering and classification are treated as methods to organise documents, and thus are helpful to access information [11]. Classification is supervised categorisation when classes are known; clustering is unsupervised categorisation when classes are not known. However, when different pre-assigned categories of documents contain

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 154–166, 2004.

many of the same features, i.e. words, it is not easy for traditional unsupervised clustering methods to organise documents based on their pre-classified categories [1].

An example of different decisions by document clustering and classification is illustrated in Fig. 1. There are nine documents which are pre-classified as two categories. Documents pre-classified as one category are represented as black circles and documents pre-classified as the other category are represented as white circles. However, based on mutual similarities of document vectors, nine documents form two clusters in Fig. 1A. The distance from document 1 to document 2 is shorter than that to document 5, so document 1 is in the same cluster as document 2 (Fig. 1B). Without embedding classification knowledge in the clustering approach, it is hard for document 1 to be grouped with document 5 (Fig. 1C).

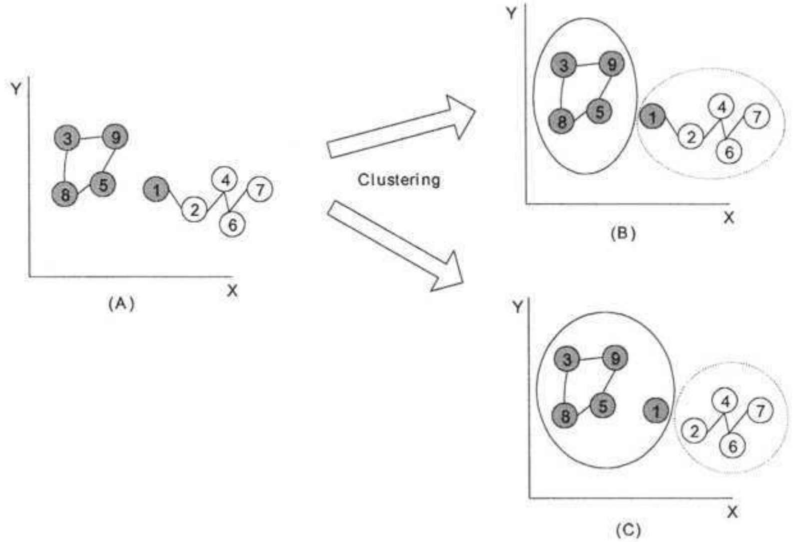


Fig. 1. An example of different decisions for document clustering and classification. Documents are represented as circles with numbers. Circles with the same filled colour are pre-assigned to the same category

Kohonen et al. [18] summarise the main purpose of neural text clustering as *"the goal is to organize a given data set into a structurefrom which the documents can be retrieved easily ... this task is differentfrom the pattern recognition task, in which the goal is to classify new items and, after learning, no attention is paid to the training data."* Thus the main aim of the Self-Organising Map (SOM) [19] is to organise the given document set. They [18] point out that *"Obviously, one should provide the different words with such weights that reflect their significance or power of discrimination between the topics."* They suggest using the vector space model (VSM) [28] to transform documents to vectors if no category information is provided. However, they also state that *"If, however, the documents have some topic classification which contains relevant information, the words can also be weighted according to their Shannon entropy over the set of document classes."* A modified VSM which includes category information is used in their WebSOM project [18, 13].

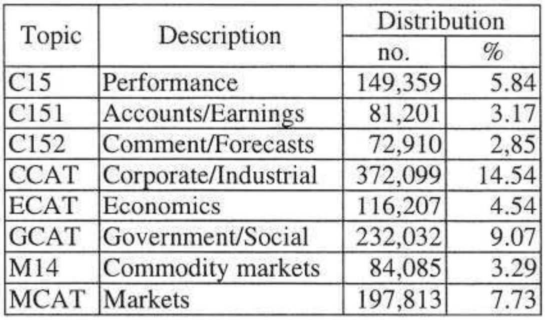
In other words, an integration of clustering and classification knowledge may take advantage from an explicit mathematical definition of clustering similarity for the classification decision and achieve a higher accuracy for clustering using classification knowledge. Consequently, a *guided* neural network based on predictive top-down classification information offers the opportunity to exploit domain knowledge, which is able to bridge the gap of inconsistency between classification knowledge and clustering decisions.

In this paper, we explore the hypothesis whether integrating linguistic top-down knowledge from WordNet [24] into a text vector representation can improve neural exploratory bottom-up clustering based on classification knowledge. By extracting semantic rules from WordNet, terms with similar concepts are substituted with a more general term. To achieve our objectives, a series of experiments will be described using several unsupervised competitive learning approaches, such as pure Competitive Learning (CL) [19, 10], Self-Organising Map (SOM) [18], Neural Gas (NG) [22], Growing Grid (GG) [8], Growing Cell Structure (GCS) [7] and Growing Neural Gas (GNG) [6]. Our experiments show that hypernyms in WordNet successfully complement neural techniques in document clustering.

2 Current Reuters Corpus of News Articles

We work with the new version of the Reuters corpus, RCV1 (It can be found at http://about.reuters.com/researchandstandards/corpus/), which consists of 806,791 news articles. There are 126 topics in this new corpus but 23 of them contain no articles. All articles except 10,186 are classified as at least one topic. In this paper, we concentrate on the most prominent 8 topics (Table 1) for our data set.

Table 1. The description of chosen topics and their distribution over the whole new Reuters corpus



Topic	Description	Distribution	
		no.	%
C15	Performance	149,359	5.84
C151	Accounts/Earnings	81,201	3.17
C152	Comment/Forecasts	72,910	2,85
CCAT	Corporate/Industrial	372,099	14.54
ECAT	Economics	116,207	4.54
GCAT	Government/Social	232,032	9.07
M14	Commodity markets	84,085	3.29
MCAT	Markets	197,813	7.73

We use the first 100,000 full-text news articles which are pre-classified according to the Reuters corpus. Because a news article can be pre-classified as more than one topic, we consider the multi-topic as a new combination topic in our task. Thus the 8 chosen topics are expanded to 54 topics (Table 2).

Table 2. The distribution of topic composition

No	Topic composition	Distribution	
		no.	%
1	ECAT/MCAT	1,034	1.03
2	CCAT	20,660	20.66
3	C15/C151/CCAT/ECAT/GCAT	32	0.03
4	C15/C151/CCAT	6,530	6.53
5	M14/MCAT	8,197	8.20
6	ECAT	7,368	7.37
7	CCAT/GCAT	3,557	3.56
8	CCAT/ECAT/GCAT	1,842	1.84
9	MCAT	11,202	11.20
10	GCAT	22,337	22.34
......			
53	C15/C151/CCAT/GCAT/M14/MCAT	1	0.00
54	C15/C151/C152/CCAT/ECAT	3	0.00
Total number of news articles		100,000	100.00

3 Extended Significance Vector Presentation

For clustering, each document must be transformed into a numeric vector. One candidate, the traditional Vector Space Model (VSM) [28] based on a bag-of-words approach is probably the most common approach. However, this model suffers from the curse of dimensionality while dealing with a large document collection because the dimensionality of document vectors is based on the total number of the different terms in the document collection. In our experiments, there are 7,223 words belonging to open-class words, i.e. nouns, verbs, adjectives, and adverbs, from 1,000 full-text news articles. In the 100,000 full-text news article task, there are 28,687 different words. Thus, some dimensionality reduction technique for a large scale document set is useful.

The most common way is leaving out the most common stop words, the most rare words and stemming a word to its base form. However, Riloff [26] suggests that these "unimportant words" will make a big difference for text classification. Only choosing the most frequent words from the whole specific word master list is also common [3]. However, there is no general heuristic to determine the threshold of the frequency. Some researchers consider this problem from a document structure viewpoint. They stress that only choosing the news headline, title, the first sentence of the first paragraph, the last sentence of the last paragraph, the first several lines or any combination above is meaningful enough for the full-text articles, e.g. [17]. However, this is decided by the information providers and therefore very subjective. Henderson et al. [12] choose so-called head of nouns and verbs using the Natural Language Processing (NLP) parser technique instead of full-text. This approach still depends on the text structure.

Another group of researchers uses vector representations and train them by clustering techniques, e.g. SOM. This cluster information from raw data is treated as input for other clustering or classification algorithms to produce a 2-stage clustering or classification model. The original version of WebSOM is one of them [13]. It consists of a word-topic SOM in its first stage and document SOM in its second stage. Pullwitt [25] proposes that the concept of a document comes from the concepts of sentences. He produces a sentence SOM and uses it to build a document SOM. Other researchers consider the dimensionality reduction problem from a mathematical view, such as Principal Component Analysis (PCA), Multi-Dimensional Scaling (MDS), Singular Value Decomposition (SVD), etc. [5]. Generally speaking, these approaches suffer from three shortcomings, which are computational complexity, information loss and difficult interpretation.

In our work, we propose another vector representation approach, which is called the extended significance vector representation. Dimensionality reduction is one major reason for using a different vector representation and another reason is the extraction of important features and filtering noise to improve the clustering performance. We do not remove common and rare words because of the evidence by Riloff [26] that these words are important. For the consistency, we restrict our experiments to those words found in WordNet, which only contains open-class words that are believed to be able to convey enough information of document concepts. The extend significance vector representation approach is started with the word-topic occurrence matrix, which is described as:

$$\begin{bmatrix} o_{11} \; o_{12} \; o_{13} \cdots\cdots o_{1M} \\ o_{21} \; o_{22} \; o_{23} \cdots\cdots o_{2M} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ o_{N1} \; o_{N2} \; o_{N3} \cdots\cdots o_{NM} \end{bmatrix}, \tag{1}$$

where $o_{ij}$is the occurrence of word i in topic j, M is the total number of topics and N is the total number of different words. An element of a significance word vector for a word i in topic j is represented as w_{ij} and is obtained using the following equation:

$$w_{ij} = \frac{o_{ij}}{\sum_{\tilde{j}=1}^{M} o_{i\tilde{j}}} . \tag{2}$$

Equation 2 can be influenced by the different number of news documents observed in each topic. When a specific topic j contains much more articles than others, a word i may contain much more occurrences in topic j than in other topics. Therefore, most words may have the same significance weights in topic j and lose the discriminatory power to topics. Equation 3 is defined as the extended significance vector, which uses the logarithmic weights of the total number of word occurrences in the data set divided by the total number of word occurrences in a specific semantic topic to alleviate skewed distributions in Equation 2. A more prominent topic which contains more word occurrences will have smaller logarithmic values. Thus, the definition of an element in a word for word i for topic j is:

$$w_{ij} = \frac{o_{ij}}{\sum_{\tilde{j}=1}^{M} o_{i\tilde{j}}} \times \log \frac{\sum_{\tilde{i}=1}^{N}\sum_{\tilde{j}=1}^{M} o_{\tilde{i}\tilde{j}}}{\sum_{\tilde{i}=1}^{N} o_{\tilde{i}j}} \tag{3}$$

Then ,the news document vector $\vec{d}$ is defined as a summation of significance word vectors $\vec{w}_i = (o_{i1}\ o_{i2}\ ......o_{iM})$ divided by the number of words in a document, which is defined as:

$$\vec{d} = \frac{1}{n}\sum \vec{w}\text{, where } n \text{ is the number of words in news document } d. \tag{4}$$

4 Extracting Top-Down Semantic Concepts from WordNet

WordNet [24] is rich of semantic relationships of synset, which is a set of synonyms representing a distinct concept. In this work, we adopt the hypernym-hyponymy relationship from WordNet to get more general concepts and thus to improve the classification ability of the SOM. A hypernym of a term is a more general term and a hyponym is more specific. We use this relationship because its gist is similar to the definition of news cluster in that the concept of a cluster of news is more general than each distinct news article. News articles with a similar concept will be grouped in the same cluster.

The vocabulary problem describes that a term can be present in several synsets. Thus, a word in different synsets may be placed in a different hypernym hierarchy (Fig. 2). It is hard to determine the right concept for an ambiguous word from several synsets and it is hard to decide the concept of a document that contains several ambiguous terms. Brezeale [2] directly uses the first synset on WordNet because of the greatest frequency of occurrence in WordNet. Voorhess [30] proposes a method called *hood* to resolve this difficulty. An ambiguous word looks for its some level hypernym until finding the same hypernym in each hypernym tree. A hood is defined as the direct descendent of this same hypernym which is shared by different concepts of a term. The meaning of ambiguous words can be decided by counting the number of other words in the text that occur in each of the different sense's hoods. Then the specific hood with the largest number is represented as the sense of ambiguous words. Scott and Matwin [29] used *hypernym density* to decide which synset is more likely than others to represent the document. The hypernym density is defined as the number of occurrences of the synset within the document divided by the number of words in the document. The synset with higher density value is more suitable to represent the document.

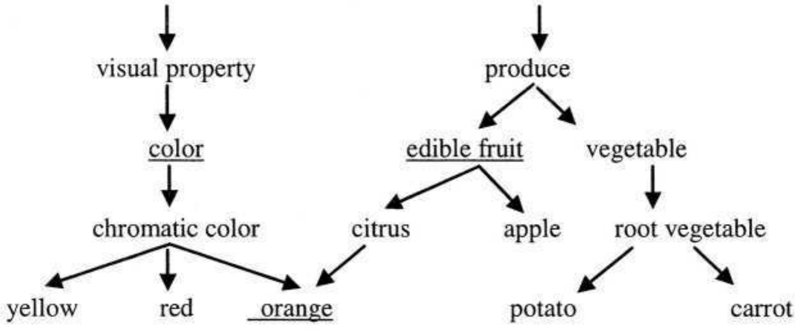


Fig. 2. An example of two hypernym trees for the term *orange*. The 2-level hypernym for orange with the colour concept is *color* but with the fruit concept is *edible fruit*

We do not use the synset directly but take advantage of the synset's gloss because two synonyms may not co-occur in a document, for example, color and colour, and orange and orangeness. The synset's gloss contains an explanation of the meaning and an example sentence of each concept. For example, the gloss of the word, *orange,* with the fruit concept is *"round yellow to orange fruit of any of several citrus trees"* and with the colour concept is *"any of a range of colors between red and yellow"*. In contrast to synonyms, words in the gloss and their target word may be more likely to co-occur.

First, we have to convert each term in our semantic lexicon into its hypernym version for every topic. We treat each gloss as a small piece of the document with a core concept and transform the gloss using the extended significance vector representation. To decide the possible gloss for an ambiguous word, the specific element weights of each gloss vector in the specific topic of the original semantic lexicon is compared. The gloss vector with the highest weights in the specific element to represent the original word is chosen. For example, a comparison is made for the first element weight only when the ambiguous word occurs in topic 1. The second element weight is compared when the ambiguous word occurs in topic 2 and so on. Then going up 2-levels in the hypernym tree, we can use this hypernym to build our hypernym version of a semantic lexicon for all terms and all categories.

To describe this approach more clearly, the following example is given. Assume that the extended significance vector of the word *orange* in the semantic lexicon is [0.234 0.033 0.502 ... 0.002] and its two gloss vectors with colour concept and with fruit concept are [0.101 0.203 0.302 ... 0.031] and [0.201 0.103 0.222 ... 0.021], respectively. When *orange* in topic 1 is converted to its hypernym, only the first element is compared for two gloss vectors. Thus, the gloss with fruit concept is chosen for *orange* in topic 1 since the first element in the gloss vector with fruit concept is greater than that with colour concept (0.201>0.101). When *orange* in topic 2 is converted to its hypernym, the colour concept is chosen (0.203>0.103). Therefore, the same word in the same topic has only one hypernym tree and different words in different topics may share the same hypernym tree (Fig. 3). Please note that to define the true meaning of an ambiguous word is not our purpose and this research rather bridges the gap of inconsistent decisions from the automated clustering technique and human classification.

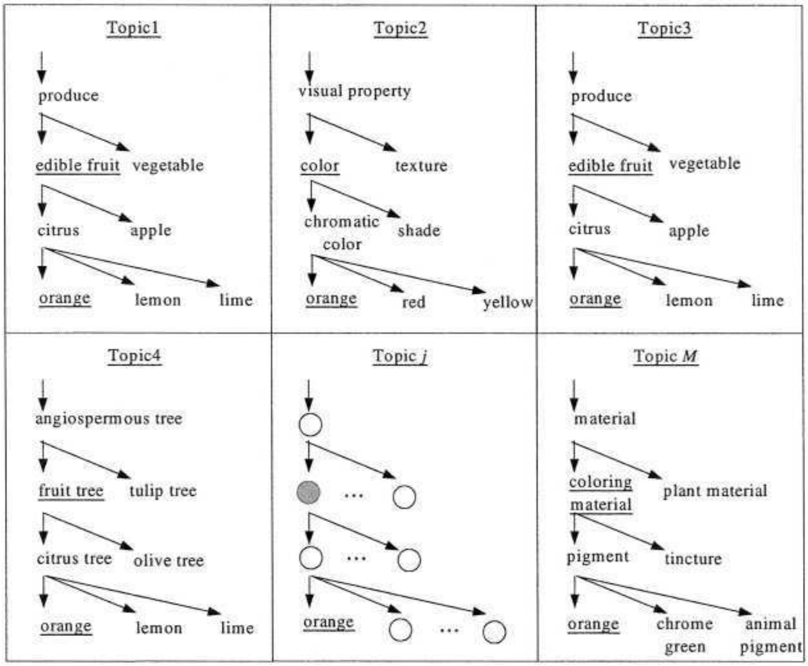


Fig. 3. An example of different hypernym trees for the term *orange*. There are four hypernym trees for this word in WordNet. Several different topics may contain the same trees, e.g. topics 1 and 3

Second, we convert each news article from its original version to its 2-level hypernym one. Because a look-up table of topic-word-hypernym has been built in stage one, we convert each word in each news document based on its classification topic and transform the 2-level hypernym data set to vectors by using the extended significance vector representation. This approach is successful to reduce the total number of distinct words from 28,687 to 11,239 for our 100,000 full-text test bed, and even improves the classification performance for several SOM-like models as we will show below.

5 Experiments with Six Competitive Learning Methods

There are several models which adopt the competitive learning principle [19, 10]. A common goal of those algorithms is to map a data set from a high-dimensional space onto a low-dimensional space, and keep its internal structure as faithful as possible. We divide these algorithms into 2 groups, i.e. static models and dynamic models, depending on whether the number of output units is static or not.

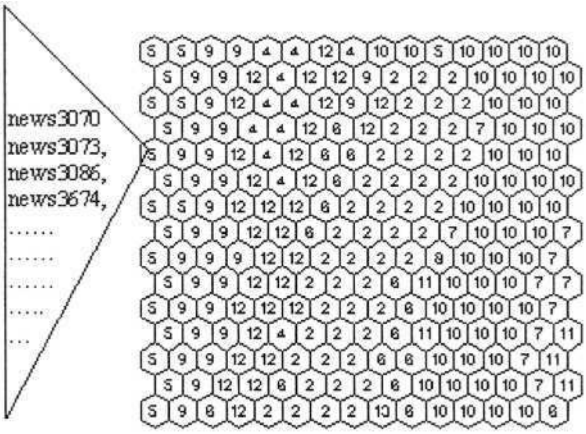


Fig. 4. The static model, SOM, with 15*15 units. Reuters topic codes are shown as numbers (Table 2)

5.1 Static Models

We test our approach with three static competitive learning models, i.e. pure Competitive Learning (CL) [10, 19], Self-Organising Map (SOM) [18] and Neural Gas (NG) [22]. The main difference between them is the way they update their cluster centres. CL is a neural version of k-means [21], which always organises its k cluster centres based on the arithmetic mean of the input vectors. CL enforces the *winner-take-all* function so only the best matching unit (BMU) of the input vector is updated. SOM is a model which mimics the self-organising feature in the brain and maps the high dimensional input into a low dimensional space, usually 2. SOM defines its own neighbouring boundary and relation in a grid. Unit centres which are inside the neighbouring boundary are updated according to the distance to the input vector. The topographic map of SOM with WordNet is shown in Fig. 4. NG is a SOM-like model without the relations between its clusters, so the clusters are treated as the gas, which can spread in the input space. All unit centres are updated based on the distance to the input vector.

5.2 Dynamic Models

Apart from the different definition of neighbourhood, dynamic models have variant dynamic representations. In this group of competitive learning algorithms, there is no need to define the number of units before training. These models will decide the number of units automatically. According to the work of Fritzke [9], a SOM model may have a good representation on the input vectors with uniform probability density but may not be suitable for complex clustering from the viewpoint of topology preservation.

In this work, Growing Grid (GG) [8], Growing Cell Structure (GCS) [7] and Growing Neural Gas (GNG) [6] are used to test our hypothesis which integrating symbolic top-down knowledge into vector representations can enhance text clustering. GG is an incremental variant of a SOM in terms of the model topology. It contains 2 stages, i.e. a growing stage, and a fine-tuning stage. Its update rule is the same in these 2 stages but the learning rate is fixed in the growing stage and is decayed in the fine-tuning stage to ensure the convergence. It starts with 2x2 units in a grid architecture which is able to keep the relative topographic relationships among units and represent input samples on a 2-dimensional map. Then GG develops the grid in column or row direction according to the position of the unit with the highest frequency of the BMU and the farthest direct neighbour of this highest BMU frequencyunit.

GCS and GNG have a unit growing feature and a unit pruning feature as well. GCS is a dynamic neural model which always keeps its units with the triangle connectivity. GCS starts with 3 units and a new unit is inserted by splitting the farthest unit from the unit with the biggest error. A unit with a very low probability density, which means few input vectors are mapped to this unit, will be removed together with its direct neighbours of the corresponding triangle. GNG is a neural model applying GCS growth mechanism for the competitive Hebbian learning topology [23].

GNG starts with 2 units and connects an input sample's BMU to the second match unit as direct neighbours. A new unit is inserted by splitting the unit with the highest error in the direct neighbourhood from the unit with the highest error in the whole structure. Units will be pruned if their connections are not strong enough. Both GCS and GNG have 2 learning rates, which are applied to BMU and BMU's direct neighbours,respectively.

5.3 A Comparison of Performance between Six Competitive Learning Models

The evaluation of SOM-like models needs more careful analysis. The unsupervised feature of SOM usually needs the inclusion of the subjective judgements of domain experts [27]. Even though it is possible to see clusters in the SOM-like maps, human qualitative judgements should not be the only evaluation criterion. The main reason is that human judgements are subjective and different assessments can be made by the same person at a different time or different process.

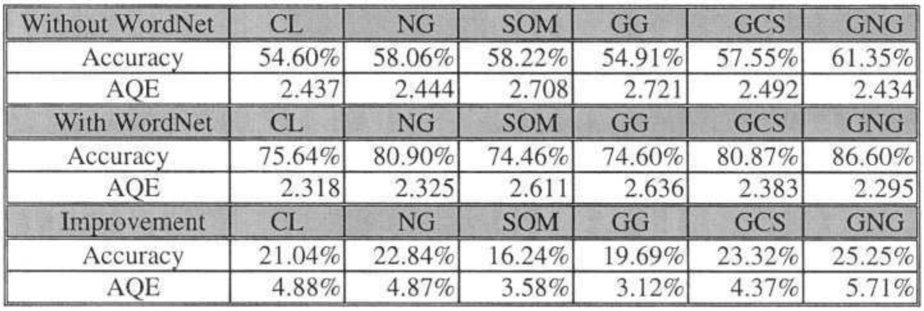
Unlike qualitative assessment, quantitative criteria or cluster validity can be divided into two types: internal and external [16]. *Internal validity* criteria are data-driven and the average quantization error (AQE) is applied in this research. The AQE tests the distortion of the representation for the model and is defined by Kohonen [20] in Equation 5. *External validity* criteria evaluate how well the clustering model matches some prior knowledge which is usually specified by humans. The most common form of such external information is human manual classification knowledge so the classification accuracy is used in this research. These two evaluation criteria have been also used by several researchers, e.g. [18, 4, 31,14, 15].

$$AQE = \frac{1}{N}\sum_{i=1}^{N}\left\|\vec{d}_i - \vec{w}_i\right\|, \text{ where } w_i \text{ is the weight vector of BMU for input sample} \tag{5}$$

i and N is the total number of input vectors.

We use 15x15 (225) units for each model and some other architectures have been tried with similar results. According to our experiments, if we use these models alone, we reach a classification accuracy between 54.60% and 61.35% for 100,000 full-text documents and an AQE between 2.721 and 2.434. Except GG, dynamic models are better in both evaluation criteria. This is because GNG and GCS contain the unit-pruning and unit-growing functions, which are able to adapt per se to input samples but GG only contains the unit-growing function and is confined its architecture to a grid, which may not reflect input samples faithfully.

Table 3. Classification accuracy and AQE without and with integration of WordNet 2-level hypernym for 100,000 full-text documents



Without WordNet	CL	NG	SOM	GG	GCS	GNG
Accuracy	54.60%	58.06%	58.22%	54.91%	57.55%	61.35%
AQE	2.437	2.444	2.708	2.721	2.492	2.434
With WordNet	CL	NG	SOM	GG	GCS	GNG
Accuracy	75.64%	80.90%	74.46%	74.60%	80.87%	86.60%
AQE	2.318	2.325	2.611	2.636	2.383	2.295
Improvement	CL	NG	SOM	GG	GCS	GNG
Accuracy	21.04%	22.84%	16.24%	19.69%	23.32%	25.25%
AQE	4.88%	4.87%	3.58%	3.12%	4.37%	5.71%

We achieve much better performance by integrating top-down knowledge from WordNet in all six algorithms based on two evaluation criteria. This hybrid approach achieves an improvement of classification accuracy from 16.24% to 25.25% and accomplishes between 74.46% and 86.60% accuracy. The AQE improvement varies from 3.12% to 5.71% and has smaller values between 2.295 and 2.636 for 100,000 full-text documents (Table 3).

6 Conclusion

In our work, we integrate symbolic top-down knowledge from WordNet into text vector representation using the extended significance vector representation technique. We examine the three static unsupervised models, Competitive Learning (CL), Neural Gas (NG) and, Self-Organizing Map (SOM) and three dynamic unsupervised models, Growing Grid (NG), Growing Cell Structure (GCS), and Growing Neural Gas (GNG) to test our hypothesis and approach. All results demonstrate that an integration of top-down symbolic information based on WordNet improves the bottom up significance vector representations in all six different approaches. Finally dynamic approaches, which determine their architecture during learning the task perform slightly better in average than static approaches. This is significant because it can avoid testing many static architectures. Our results demonstrate that our hybrid and dynamic neural model has a large potential for learning automatic text clustering.

References

1. Aggarwal, C.C., Gates, S.C., Yu, P.S.: On the Merits of Building Categorization Systems by Supervised Clustering. Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (1999) 352-356
2. Brezeale, D.: The Organization of Internet Web Pages Using WordNet and Self-Organizing Maps. Masters thesis, University of Texas at Arlington (1999)
3. Chen, H, Schuffels, C., Orwig, R.: Internet Categorization and Search: a Self-Organizing Approach. Journal of Visual Communication and Image Representation, Vol. 7. No. 1 (1996) 88-102
4. Choudhary, R., Bhattacharyya, P.: Text Clustering Using Semantics. The 11th International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA (2002)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, second edition. John Wiley & Sons, A Wiley-Interscience Publication (2001).
6. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. In: Tesauro, G, Touretzky, D.S., Leen, T.K. (eds.): Advances in Neural Information Processing Systems 7, MIT Press, Cambridge MA (1995) 625-632
7. Fritzke, B.: Growing Cell Structures – a Self-Organizing Network for Unsupervised and Supervised Learning. Neural Networks, Vol. 7, No. 9 (1994) 1441-1460
8. Fritzke, B.: Growing Grid-a Self-Organizing Network with Constant Neighborhood Range and Adaptation Strength. Neural Processing Letters. Vol. 2, No. 5 (1995) 9-13
9. Fritzke, B.: Kohonen Feature Maps and Growing Cell Structures – a Performance Comparison. In: Giles, C.L, Hanson, S.J., Cowan, J.D. (eds.): Neural Information Processing Systems 5. Morgan Kaufmann, San Meteo, CA (1993)
10. Grossberg, S.: Competitive Learning: from Interactive Activation to Adaptive Resonance. Cognitive Science, vol. 11 (1987) 23-63
11. Hearst, M. A.: The Use of Categorise and Clusters for Organizing Retrieval Results. In: Strzalkowski, T. (ed.): Natural Language Information Retrieval, Kluwer Academic Publishers, Netherlands (1999) 333-374
12. Henderson, J., Merlo, P., Petroff, I. and Schneider, G.: Using Syntactic Analysis to Increase Efficiency in Visualizing Text Collections. 19th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan (2002) 335-341
13. Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: Exploration of Full-Text Databases with Self-Organizing Maps. Proceedings of the International Conference on Neural Networks (ICNN'96), Washington (1996) 56-61
14. Hung, C. and Wermter, S.: A Self-Organising Hybrid Model for Dynamic Text Clustering. Proceedings of The Twenty-third SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2003), Cambridge, UK (2003)
15. Hung, C., Wermter, S.: A Dynamic Adaptive Self-Organising Hybrid Model for Text Clustering. Proceedings of The Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, USA (2003)
16. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ (1988)
17. Ko, Y., Seo, J.: Text Categorization Using Feature Projections. 19th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan (2002)
18. Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., Saarela, A.: Self Organization of a Massive Document Collection. IEEE Transactions on Neural Networks, Vol. 11, No. 3 (2000) 574-585
19. Kohonen, T.: Self-Organization and Associative Memory. Springer-Verlag, Berlin (1984)
20. Kohonen, T.: Self-Organizing Maps. 3rd edition. Springer-Verlag, Berline, Heidelberg, New York (2001)

21. MacQueen, J.: On Convergence of K-Means and Partitions with Minimum Average Variance. Ann. Math. Statist., Vol. 36 (1965) 1084
22. Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: Neural-Gas Network for Vector Quantization and Its Application to Time-Series Predication. IEEE Transactions on Neural Networks, Vol. 4, No. 4 (1993) 558-569
23. Martinetz, T.M.: Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps. International Conference on Artificial Neural Networks, ICANN'93, Amsterdam (1993) 427-434
24. Miller, G.A.: WordNet: a Dictionary Browser. Proceedings of the First International Conference on Information in Data, University of Waterloo, Waterloo (1985)
25. Pullwitt, D.: Integrating Contextual Information to Enhance SOM-Based Text Document Clustering. Neural Networks, Vol. 15 (2002) 1099-1106
26. Riloff, E.: Little Words Can Make a Big Difference for Text Classification. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1995) 130-136
27. Roussinov, D.G., Chen, H.: Document Clustering for Electronic Meetings: an Experimental Comparison of Two Techniques. Decision Support Systems, Vol. 27, (1999) 67-79
28. Salton, G.: Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley. USA (1989)
29. Scott, S., Matwin, S.: Text Classification Using WordNet Hypernyms. Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems (1998) 38-44
30. Voorhees, E.M.: Using WordNet to Disambiguate Word Senses for Text Retrieval. Proceedings of the Sixteenth Annual International ACM SIGIR conference on Research and Development in Information Retrieval (1993) 171–180
31. Wermter, S., Hung, C.: Selforganising Classification on the Reuters News Corpus. The 19th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan (2002) 1086-1092

Contextual Document Clustering

Vladimir Dobrynin[1], David Patterson[2], and Niall Rooney[2]

[1] Faculty of Applied Mathematics and Control Processes,
St Petersburg State University, 198904 Petrodvoretz, St. Petersburg, Russia
vdobr@oasis.apmath.spbu.ru
[2] Nikel, Faculty of Informatics, University of Ulster
{wd.patterson,nf.rooney}@ulster.ac.uk

Abstract. In this paper we present a novel algorithm for document clustering. This approach is based on distributional clustering where subject related words, which have a narrow context, are identified to form metatags for that subject. These contextual words form the basis for creating thematic clusters of documents. In a similar fashion to other research papers on document clustering, we analyze the quality of this approach with respect to document categorization problems and show it to outperform the information theoretic method of sequential information bottleneck.

1 Introduction

Document clustering is an important feature of information retrieval systems and can be used to solve many varied tasks, from the search for duplicate documents to the visualization of retrieval results and web mining [22]. In this paper we present a novel approach to clustering documents based on word distributions, so that each cluster is identified by a context word and the documents contained in the cluster have a high level of specificity and relevancy to each other. We propose this technique as a means of indexing a document store and show its efficacy through a series of document classification experiments.

Generally clustering algorithms can be categorized as hierarchical (agglomerative and divisive) or partitional in nature [9]. Partitional clustering such as the well known k-means tends to be a much more efficient approach to clustering, although it in general requires apriori knowledge of the number of clusters. Most clustering algorithms determine the similarity between points based on a distance measure. This forms the basis of standard clustering approaches such as k-means, k-medoids and hierarchical methods such as single-link, complete link and group average clustering [9]. Recent extensions to these classical approaches have included clustering by committees [15] which uses the mutual information between documents based on their respective feature vectors to determine similarity. Committees of clusters are then formed where each member covers a tight number of closely related clusters and each committee member is defined to be dissimilar to others. This approach is shown to outperform many of the classical approaches. Another approach proposed by Liu et al [14] uses a richer feature set to represent each document. Clustering is then performed using a

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 167–180, 2004.

Gaussian mixture model together with an Expectation-Maximization algorithm which iteratively refines the clusters based on an evaluation of which features are discriminatory.

In terms of clustering a corpus of documents, a more natural measure of the similarity between documents is based on the word distributions of the documents. An information theoretic framework can then be applied to cluster documents or words/features in the documents. Such approaches belong to the area of distributional clustering [2,16] where the similarity measure is based on information theoretical divergence crtieria [13]. The goal of word/feature clustering is to provide a mechanism of feature reduction, where the original feature space is transformed into the space of new features represented by the word clusters. Feature reduction is a desirable way of addressing the problem of high feature dimensionality within document corpora, and as such feature clustering is a more effective mechanism than feature selection [2,6].

The most recent focus on the research in distributional clustering has focussed primarily on the Information Bottleneck (IB) clustering framework [20]. The IB method is based on the following abstract concepts. Given the empirical joint probability distribution of two variables, one variable is "compressed" so that the maximum mutual information is maintained between both. In the context of clustering, the variables represent the set of documents and the set of words. Using this method, it is possible to form either word clusters or document clusters. In terms of document clustering, the original IB algorithm was agglomerative and sub-optimal in nature (i.e. does not necessarily form the "best" clusters). A number of new algorithms have been presented based on IB, with the intent of either trying to improve its performance in terms of text categorization problems or its efficiency or both [3,4,7,18,19]. For example the sequential informational Bottleneck (siB) [19] method is partitional which improves its efficiency and ensures that the most optimal partition of documents into clusters is formed given the mutual information.

While the approach we present in this paper is based on distributional clustering, our research goals differ from other distributional document clustering such as IB. The later tries to provide a more compact representation of the data, whereas we are primarily interested in identifying documents which belong to highly specific contexts, for example a document collection may contain the word "computer" and "motherboard", but it is likely that the former word will occur in a diverse range of documents with no subject relation whereas the latter word will occur in documents that are highly related. Such words we consider as having a "narrow" context. In essence these contextual words provide a mechanism of grouping together semantically related documents. This Contextual Document Clustering (CDC) approach of identifying words that form clusters of narrow scope will lend itself well to organize a collection of documents in a more meaningful fashion. It is worth stressing also that the contextual words are not identified by any pre-defined categories that the documents belong to, but are determined automatically in a unsupervised fashion. It is important that a document clustering algorithm can operate in a unsupervised manner because for many document corpora, such predefined categories will not exist or will be unknown. CDC is more generic than other approaches in that it does not require such prior knowledge. The motivation behind CDC is our belief that an information retrieval system can be built in a more effective manner by grouping documents into clusters using CDC rather than standard indexing approaches, such as Latent semantic indexing which due to the transformation of the data space are inevitably going to result in the loss of valuable information. The efficacy of CDC is measured by how many relevant documents are retrieved as a result of a query (*the precision of the mechanism*), and how many relevant documents in the corpus are actually returned *(the recall)*. CDC should allow the retrieval process for a given query to be performed efficiently as it only has to identify initially the small number of clusters relevant to the query. These clusters will contain on average a small number of documents relative to the corpus size as a whole. In early research in Information retrieval, document clustering was considered as a desirable method of indexing documents [21] but the approach was not pursued due to the high computational demands required at the time. As such recent research on document clustering has tended to focus on its applicability to clustering the results of a query in an IR system. Document categorization has been used frequently in the literature as an independent measure of determining the quality of the clustering process. Typically document categorization is a classification problem where a document is represented using the bag of words model, some form of feature reduction is performed and a multi-class, multi-labelled categorization is carried out using a standard classifier such as SVM or Naive Bayes. A review of such techniques is presented in [17]. Although this approach provides the best known results for well known data-sets such as the Reuters-21578, it suffers from the practical issue that the bag-of-words may have high dimensionality even after feature reduction. In our evaluation section we compare the results of document categorization performed on the Reuters-21578 and 20Newsgroups for the CDC technique and discuss them in comparison to the recent document clustering approach, (sIB) [19].

2 Contextual Clustering

In this Section we provide a theoretical description of CDC. In addition, we provide an analysis of the algorithm's time complexity, and summarize the main aspects concerning sIB, as we compare our results to this technique. Contextual Clustering consists of two steps, identifying the narrow contexts and clustering the documents where contexts act as cluster centroids.

2.1 Context Word Identification

The term "word context" is used to describe the probability distribution of a set of words which co-occur with the given word in a document. In other words the word context is described by a conditional probability distribution

$$p(Y|z),$$

where Y is a random variable with values in the collection dictionary and z is the given word which acts as a descriptor of the context.

As described in the introduction, it is obvious that not all word contexts are useful only those that have a narrow context. Let $\mathcal{X}$ be the set of all documents in the document collection and $\mathcal{Y}$ be set of all words occurring in documents from $\mathcal{X}$. The context "narrowness"is measured by a determination of the entropy of its probability distribution and the document frequency for a context word z.

Let $tf(x,y)$ denote the number of occurrence of the word $y \in \mathcal{Y}$ in the document $x \in \mathcal{X}$. Then the empirical joint probability distribution $p(X,Y)$ of random variables X and Y with values from $\mathcal{X}$ and $\mathcal{Y}$ is calculated by

$$p(x,y) = \frac{tf(x,y)}{\sum_{x',y'} tf(x',y')},$$

where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ and the independence of word occurrences in the collection documents is assumed, so that

$$p(y_1,y_2|x) = p(y_1|x)p(y_2|x),$$

where $y_1,y_2 \in \mathcal{Y}$and $p(y_1,y_2|x)$denote the probability of co-occurrence of words y_1 and y_2 in the document x.

Words with a narrow word contexts are selected by calculating the conditional probability distributions $p(Y|z)$ for all word $z \in \mathcal{Y}$:

$$p(y|z) = \frac{p(y,z)}{p(z)}$$

$$p(y,z) = \sum_x p(x)p(y|x)p(z|x),$$

$$p(z) = \sum_x p(x,z),$$

$$p(x) = \sum_y p(x,y)$$

then the empirical probability distribution for $p(y|z)$ is:

$$p(y|z) = \sum_{x \in D_z} \frac{tf(x,y)}{\sum_{y'} tf(x,y')} \cdot \frac{tf(x,z)}{\sum_{x' \in D_z} tf(x',z)}, \tag{1a}$$

where $D_z$is the set of all documents with word z.

This is equivalent to the probability that word y is chosen given that a document is randomly chosen from D_z (with probability proportional to the number of occcurences of word z in the document).

and measuring the entropy:

$$H(Y|z) = H[p(Y|z)] = -\sum_y p(y|z) \log p(y|z).$$

The above approach to calculate $p(y|z)$ suffers from the disadvantage that the function $p(x, y)$ must be re-calculated any time a new document is added to the document corpus. A more amenable approach to incremental learning could be determined by assuming that $p(x)$ is the same for all documents and by calculating the empirical conditional distribution $p(y|z)$ as:

$$p(y|z) = \frac{\sum_{x \in D_z} tf(x, y)}{\sum_{x \in D_z,\ y'} tf(x, y')}, \tag{1b}$$

where the assumption is not made that the word occurrences in a document are independent. (1b) is equivalent to randomly choosing a word y given that a document from D_z has been selected with probability proportional to its length.

Let (1a) denote narrow word selection criterion when the independence of word occurrences in a document is assumed and (1b) otherwise. Both (1a) and (1b) are unsupervised techniques.

Only those word contexts are selected as narrow based on a consideration of the entropy $H(Y|z)$ and the document frequency $df(z) = |\{x : tf(x, z) > 0\}|$. The maximum entropy $H_{max}(Y|z)$ occurs for a uniform distribution and equals the log $|T(D_z)|$ where $T(D_z)$ is the set of words belonging to documents where z occurs. Heaps Law [1] states that the dictionary size of a document collection is of the order $O(n^{\beta})$, where n is the total text size and $\beta < 1$. As the text size for documents where z occurs is $df(z)$ times a constant k denoting the average size for these documents, then $|T(D_z)| = O((k \cdot df(z))^{\beta})$ and $H(Y|z) = O(\log df(z))$

To take into account the dependence between $H(Y|z)$ and $df(z)$ the set of words $\mathcal{Y}$ is divided into a set of subsets such that all words from a given subset have document frequencies from the same interval:

$$\mathcal{Y} = \cup_i \mathcal{Y}_i,$$

$$\mathcal{Y}_i = \{z : z \in \mathcal{Y},\ df_i \le df(z) < df_{i+1}\},$$

$$i = 1, \ldots, r.$$

If

$$df_{i+1} = \alpha \cdot df_i,$$

where $\alpha > 1$ is a constant, then $H_{max}(Y|z)$ is bounded from above by a linear function of the interval $[df_i,\ df_{i+1})$ index i. So as a consequence a threshold $H_{max}(i)$ is set for every interval $[df_i,\ df_{i+1})$ and select word set $\mathcal{Z} \subset \mathcal{Y}$ such that

$$\mathcal{Z} = \cup_i \{z : z \in \mathcal{Y},\ df_i \le df(z) < df_{i+1},$$

$$H(Y|z) \le H_{max}(i)\}. \tag{2}$$

The entropy threshold values are selected empirically based on the entropy distribution over all possible contexts for given a document corpus. As this entropy distribution is dependent on the approach used to calculate the conditional probabilities $p(y|z)$, different threshold values may be selected for criterion (1a) than those for criterion (1b) for a given document corpus. An additional constraint is included in Criterion (1b) to limit the maximum number of words W_c contained within the contexts where naturally only those words y with the maximum conditional probability $p(y|z)$ are kept. This was added also to improve the efficiency of (1b) over (1a) in the situation where a context or contexts contain a large number of words > W_c. It also may improve the accuracy of the technique by filtering out irrelevant words.

An alternative supervised means of choosing the narrow word contexts is to assume that in total there are N word contexts and r document frequency intervals as before. If $\mathcal{Y}_i$ denotes the set of words from the corpus dictionary which contains all words z such that $df_i \le df(z) < df_{i+1}$. For every $i = 1, \dots, r$ a set $\mathcal{Z}_i \subset \mathcal{Y}_i$ is selected such that :

$$|\mathcal{Z}_i| = \frac{N \cdot |\mathcal{Y}_i|}{\sum_{j=1,\dots,r;} |\mathcal{Y}_j|}$$

and

$$z_1 \in \mathcal{Z}_i,\ z_2 \in \mathcal{Y}_i - \mathcal{Z}_i \rightarrow H(Y|z_1) \le H(Y|z_2). \tag{3}$$

Then $\mathcal{Z} = \cup_i \mathcal{Z}_i$. where $\mathcal{Z}$ is the set of selected word contexts. This second approach has the advantage that the entropy threshold bounds do not need to be determined empirically for a particular data-set.

Criterion (2) is similar to criterion (1b) in that the distribution $p(y|z)$ is calculated directly and there is a limit also on the maximum number of words a context may contain.

2.2 Document Clustering

Selected narrow word contexts act as cluster centroids and the clustering process works by assigning every document from the document collection to the cluster with the closest centroid, as measured by the JS-divergence.

Let x be a document. Then conditional probability distribution of its words $p(Y|x)$ is given by

$$p(y|x) = \frac{p(x,y)}{p(x)},$$

and the distance between document x and word z context is measured using the Jensen-Shannon divergence [13] of $p(Y|z)$ and $p(Y|x)$ distributions.

Let $p_1(Y)$ and $p_2(Y)$ be two probability distributions of a random variable Y. Then the Jensen-Shannon divergence of p_1 and p_2

$$JS_{\{\pi_1,\pi_2\}}[p_1,p_2] = H[\bar{p}] - \pi_1 H[p_1] - \pi_2 H[p_2],$$

where $\pi_1 \ge 0,\ \pi_2 \ge 0,\ \pi_1 + \pi_2 = 1,\ \bar{p} = \pi_1 p_1 + \pi_2 p_2$, is a nonnegative bounded function of p_1 and p_2 which is equal to zero iff $p_1 = p_2$. Also this function is concave with respect to π_1 and π_2 with unique maximum value in the point {0.5,0.5}.

So the document x will be assigned to the cluster C_z with centroid for word z if

$$z = \arg\min_{z'} JS_{\{0.5,0.5\}}[p(Y|z'), p(Y|x)].$$

2.3 Complexity Analysis of CDC

Let S be a set of documents used to generate N contexts, $df_{avr}(S)$ the average document frequency of a word in S, $length_{avr}(S)$ average document length in S, and $T(S)$ the set of all distinct words in S, then the time complexity for context generation is

$$O(|T(S)| \cdot df_{avr}(S) \cdot length_{avr}(S)).$$

If $length_{avr}(S)$ is bounded from above by a constant value not dependent on $|S|$ then time complexity for context generation is

$$O(|S|)$$

and time complexity for clustering is

$$O(N \cdot |S|).$$

From this it clear that the process of the context word discovery and document clustering is only linearly dependent on the size of the document collection, so it is much more efficient mechanism than standard document clustering approaches which are usually quadratic in their time complexity [5].

2.4 Sequential Information Bottleneck

In this section we review the sIB technique. As stated given a joint distribution $p(X|Y)$, the Information Bottleneck method looks for a compact representation of X which preserves as much information as possible about the relevant variable Y. The mutual information, $I(X; Y)$, between the random variables X and Y is given by:

$$I(X;Y) = -\sum_{x \in X, y \in Y} p(x)p(y|x) \log \frac{p(y|x)}{p(y)}$$

A compressed representation T of X is defined by $P(T|X)$. The compactness of the representation is determined by $I(T; X)$ while the quality of the clusters is measured by the fraction of information they capture about Y, namely $I(T;Y)/I(X;Y)$. Intuitively in this procedure the information contained in X about Y is "squeezed" through a compact "bottleneck" of clusters T that is forced to represent the relevant part of X with respect to Y. The sequential clustering works by finding a partition of $T(X)$ which maximizes a score function given by $I(T;Y)$.The algorithm starts with an initial random partition $T = \{t_1, t_2, .., t_K\}$ of X. Similar to the k-means algorithm K is a pre-determined value. At each step in the method some $x \in X$ is drawn out of its current cluster

$t(x)$ and is represented as a new singleton cluster $\{x\}$. x is then merged into t^{new} such that $t^{new} = argmin_{t \in T} d(\{x\}, t)$ where d is

$$d(x,t) = (p(x) + p(t)) JS(p(y|x), p(y|t))$$

Slonim [19] shows that this iterative process is guaranteed to converge to a local maximum of the score function, where no more re-assignments can improve upon the result.

3 Experimental Setup

The experimental evaluation of the contextual clustering technique for document categorization used two standard data-sets.

The first dataset was the training and test parts of the ModApte split of the Reuters-21578 collection where the documents selected fall into 10 most frequent categories ("earn", "acq", "money-fx", "crude", "grain", "trade", "interest", "wheat", "ship", "corn"). Note that because documents can be assigned to more than one category, the total number of categories documents can belong to is 102. Note that these categories are not used to cluster documents contextually. The cluster context's are determined as previously described in section 2.1 in a totally unsupervised fashion. However this domain knowledge about the document categories is used to assess the quality of the CDC clustering.

Document preprocessing includes

- removing all file headers except *title* and *dateline*
- lowering upper case characters
- removing all digits and all non alpha-numeric characters
- removing stop words (470 words) and words which occur in the collection only once
- removing the body of short documents (all information such document contains is stored in its title)

As a result there were 9035 documents in this collection and 16720 unique words.

To select a word (z) with narrow word context the bounds for criteria (1a) were empirically set for this data-set to:

$$5 \leq df(z) < 12 \quad AND \quad H(Y|z) \leq 5.00 \quad OR$$

$$12 \leq df(z) < 25 \quad AND \quad H(Y|z) \leq 5.25 \quad OR$$

$$25 \leq df(z) < 50 \quad AND \quad H(Y|z) \leq 5.50 \quad OR$$

$$50 \leq df(z) < 100 \quad AND \quad H(Y|z) \leq 5.75 \quad OR$$

$$100 \leq df(z) \quad AND \quad H(Y|z) \leq 6.0.$$

This resulted in 907 narrow word contexts.

To select a word (z) with narrow word context the bounds for criterion (1b) were empirically set for this data-set to:

$$5 \leq df(z) < 12 \;\; AND \;\; H(Y|z) \leq 5.25 \;\; OR$$
$$12 \leq df(z) < 25 \;\; AND \;\; H(Y|z) \leq 5.50 \;\; OR$$
$$25 \leq df(z) < 50 \;\; AND \;\; H(Y|z) \leq 5.75 \;\; OR$$
$$50 \leq df(z) < 100 \;\; AND \;\; H(Y|z) \leq 6.0 \;\; OR$$
$$100 \leq df(z) \;\; AND \;\; H(Y|z) \leq 6.25.$$

and W_c was set to 1000. This resulted in 854 narrow word contexts.

Using criteria (2), the document frequency intervals were set to $df_1 = 5$, $df_2 = 12$, $df_3 = 25$, $df_4 = 50$, $df_5 = 100$, $df_6 = |corpus|$ and the number of contexts was set $N = 1000$, (as criteria (1) had resulted in 907 word contexts). Naturally this results in a 1000 narrow word contexts.

The second data-set was the 20 Usenet Newsgroups (20NG) as collected by [12]. This corpus contains roughly 20000 documents evenly distributed over 20 newsgroups, some of which are of very similar topics. There is roughly 4% cross-posting in this corpus. This data-set was filtered to remove duplicate documents Additional preprocessing included

- removing all headers except *subject*
- lowering upper case characters
- removing all non alpha-numeric characters and tokens without alphabetic symbols
- removing stop words (470 words) and words which occur in the collection only once

As a result there were 18938 documents in this collection and 58879 unique words.

To select the set $\mathcal{Z}$ of words which mark narrow contexts for this collection we used only criterion (1b) and criterion (2) (Criterion (1a) proved too inefficient for this data-set).

Criterion (1b) was set with empirical bounds

$$5 \leq df(z) < 12 \;\; AND \;\; H(Y|z) \leq 5.50 \;\; OR$$
$$12 \leq df(z) < 25 \;\; AND \;\; H(Y|z) \leq 6.00 \;\; OR$$
$$25 \leq df(z) < 50 \;\; AND \;\; H(Y|z) \leq 6.50 \;\; OR$$
$$50 \leq df(z) < 100 \;\; AND \;\; H(Y|z) \leq 7.0 \;\; OR$$
$$100 \leq df(z) \;\; AND \;\; H(Y|z) \leq 7.5.$$

and W_c was set to 1000 as before. This resulted in 1422 contexts being selected. The 20NGs were investigated with the same settings as the Reuters data for criterion(2).

4 Precision and Recall

In this section, we describe how precision and recall are measured in the experiments. Let document x_z be the nearest document to the centroid (context of the word z) of the cluster C_z and $T_e(x_z)$ be the category set of document x_z assigned by an expert (the predefined cateogories). Then the category set $T(C_z)$ is assigned to the cluster C_z and has value

$$T(C_z) = T_e(x_z)$$

if

$$x_z = \arg\min_{x'} JS_{\{0.5,0.5\}}[p(Y|z), p(Y|x')].$$

All documents, assigned to the cluster C_z, are classified as category set $T(C_z)$, whereas the true classification of a document x is $T_e(x)$. A document $x \in corpus$ is considered to be classified correctly if the intersect of $T_e(x)$ and the category set it is assigned to $T(C(x))$ is non-empty:

$$T(C(x)) \cap T_e(x) \neq \emptyset. \tag{4}$$

Then the precision p_1 of the classification is calculated as

$$p_1 = \frac{|\{x : T(C(x)) \cap T_e(x) \neq \emptyset\}|}{|corpus|}.$$

The quality of the CDC method can be calculated in a more stringent fashion. Let T be the set of all categories, $t \in T$ and given the following definitions:

- Number of "true positives" for the category t

$$TP_t = |\{x : x \in C_z \rightarrow t \in T(C_z) \cap T_e(x)\}|$$

- Number of "false positives" for the category t

$$FP_t = |\{x : x \in C_z \rightarrow t \in T(C_z) - T_e(x)\}|$$

- Number of "false negatives" for the category t

$$FN_t = |\{x : x \in C_z \rightarrow t \in T_e(x) - T(C_z)\}|$$

Then the microaverage precision p^μ is calculated as

$$p^\mu = \frac{\sum_{t \in T} TP_t}{\sum_{t \in T}(TP_t + FP_t)}$$

and microaverage recall r^μ as

$$r^\mu = \frac{\sum_{t \in T} TP_t}{\sum_{t \in T}(TP_t + FN_t)}$$

In order to compare CDC to the sIB technique described in [19], we also present results based on the latter paper's measurements of precision and recall. In the sIB evaluation approach, all documents are assigned to the most dominant label in that cluster. A document is considered correctly classified if its true label set contains the dominant label. These dominant labels form a category set C and for each $c \in C$ is measured $\alpha(c)$, the number of documents correctly assigned to c; $\beta(c)$, the number of documents incorrectly classified to c and $\gamma(c)$, the number of documents incorrectly not assigned to c. Then the precision and recall are defined as:

$$p_{sIB} = \frac{\sum_{c \in C} \alpha(c)}{\sum_{c \in C}(\alpha(c) + \beta(c))}$$

$$r_{sIB} = \frac{\sum_{c \in C} \alpha(c)}{\sum_{c \in C}(\alpha(c) + \gamma(c))}$$

In general this tends to give a more optimistic measure of precision and recall than that defined by p^{μ} and r^{μ}.

5 Experimental Results

5.1 Reuters Dataset

The precision and recall results for the Reuters data-set are presented in Table 1 for the three narrow word selection criteria. Results are shown both for the case where documents closest to the context centroids are kept and when they are removed shown in brackets in Table 1. The reason for their exclusion it that they may be considered to bias optimistically the results as their category sets are used to measure precision.

Table 1. Classification Results for the Reuters data-set using both criteria

Criteria	$1(a)$	$1(b)$	2
number of contexts	907(907)	854(854)	1000(1000)
number of non-empty clusters	863(848)	790(781)	893(879)
average cluster size	10.5(10.0)	11.4(10.9)	10.1(9.7)
max cluster size	448(447)	458(458)	459(456)
p_{sIB}	0.886	0.886	0.888
p_1	0.832(0.822)	0.845(0.836)	0.845(0.836)
p^{μ}	0.742(0.725)	0.697(0.681)	0.69(0.673)
r^{μ}	0.76(0.744)	0.765(0.75)	0.773(0.758)

It should be observed that each criterion resulted in a high number of non-empty clusters which contain on average a small number of documents. There was not much variation in the number of clusters, average cluster size and maximum cluster size among the criteria. p_{sIB} gave the highest value as a measure of precision as expected (0.89). This outperformed sIB which for this data-set had a precision value of 0.86 [19] (a 3% increase). It also significantly outperformed sIB in terms of recall. sIB is a hard clustering technique which tries to constrain the Reuters data-set to the same number of clusters as there are most frequent categories i.e. 10. However as this data-set is multi-labelled, its recall was low (at best 0.56 [19]), which is 20% lower than CDC.

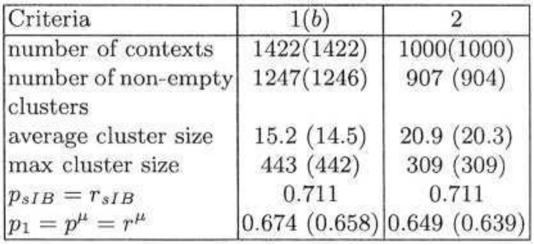
The values of precision (regardless of the method of measurement) and recall are very similar to each other whichever criteria we used, showing that each criterion was equally valid in identifying narrow word contexts. However criterion (1a) was a much slower technique than the other two. As would be expected, removing the closest documents to the centroid results in a slight reduction in precision and recall which is at worse 1.7%.

5.2 20Newsgroups Dataset

The results for precision and recall for the 20Newsgroups data-set are presented in Table 2. Results are shown both for the case where documents closest to the context centroids are kept and when they are removed in a similar fashion to Table 1.

Note that this data-set is uni-labelled. As a result the micro-average precision p^{μ}, micro-average recall r^{μ} and p_1 have the same value. Similarly p_{sIB} is the same as r_{sIB} So only the value for micro-average precision is presented in the Table 2. Again as was the case for the Reuters data, there was a high number of non-empty clusters and a small overall average cluster size.

Table 2. Results for 20NGs data-set



Criteria	$1(b)$	2
number of contexts	1422(1422)	1000(1000)
number of non-empty clusters	1247(1246)	907 (904)
average cluster size	15.2 (14.5)	20.9 (20.3)
max cluster size	443 (442)	309 (309)
$p_{sIB} = r_{sIB}$	0.711	0.711
$p_1 = p^{\mu} = r^{\mu}$	0.674 (0.658)	0.649 (0.639)

Table 2 shows that both criteria resulted in a high number of non-empty clusters with a small average cluster size. The precision p_{sIB} was the same for both criterion (0.71) and significantly outperformed sIB which had a value of precision and recall, of 0.58 [19], 13% less than CDC's p_{sIB}. Again $p_1 = p^{\mu} = r^{\mu}$ were very similar for both criteria, indicating the validity of both criteria for contextual clustering. More significance should be put on the results for the 20NGS than for the Reuters data-set. The reason for this is that the Reuters data-set is quite simplistic. In other words, it is a relatively easy for a categorization technique to achieve high values of precision [17]. The 20NGs data-set is a more challenging data-set and therefore it is more difficult to obtain such high levels of precision. As such the results give strength to our belief that CDC is a competent technique for document clustering.

In summary, CDC provided a more competent technique for document categorization than the sIB approach. Criteria 1(a—b) and 2 were equally accurate in terms of precision and recall, however criterion 1(b) and 2 provided a more efficient means of clustering the documents. The results demonstrate emphatically the benefit of forming a relatively high number of small contextual clusters.

6 Conclusion

The contextual clustering approach based on narrow context word selection partitions a document collection into a large number of relatively small thematic homogeneous clusters, regardless of the clustering criteria used. The number of clusters generated by CDC is a reflection of the real complex thematic structure of a document corpus, which can not be adequately expressed by grouping documents into a small number of categories or topics. This was borne out in the experimental evaluation of CDC. It showed a high precision and recall, when applied to the document categorization of the ModApte split Reuters data and the 20NG data-sets, and which outperformed another information theoretic document clustering approach, sIB which relies on the user-defined categories.

References

1. Baeza-Yates and Ribeiro-Neto: Modern Information Retrieval, ACM Press, 1999.
2. Baker, L.D., McCallum, A.K.: Distributional clustering of words for text classification. In Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval, pp. 96-103, 1998.
3. Bekkerman, R., El-Yaniv, R., Tishby, N., Winter, Y.: On feature distributional clustering for text categorization. In Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval, pp. 146-153,2001.
4. Bekkerman, R., El-Yaniv, R., Tishby, N., Winter,Y.: Distributional word clusters vs. words for text categorization. Journal of Machine Learning Research, Vol 1:1-48, 2002.
5. Cutting, D.,Pedersen, J., Karger, D., Tukey, J.: Scatter/Gather: Cluster-based Approach to Browsing Large Document Collections. In Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 318-329, 1992.
6. Dhillon, Y.,Manella, S., Kumar, R.: Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification, Journal of Machine Learning Research Vol 3:1265-1287, 2003.
7. El-Yaniv R., Souroujon O.: Iterative double clustering for unsupervised and semi-supervised learning. In Proceedings of ECML-01, 12th European Conference on Machine Learning. pp. 121 - 132,2001.

8. Hofmann, T.: Probabilistic latent semantic indexing. In Proceedings of the 22nd ACM-SIGIR Intemational Conference on Research and Development in Information Retrieval, pp. 50-57, 1999.
9. Jain, A. K., Murty, M. N. and Flynn, P. J.: Data Clustering: A Review. ACM Computing Surveys 31(3):26423,1999.
10. Joachims, T.: A statistical learning model for Support Vector Machines. SIGIR'01, New Orleans, USA, 2001.
11. Karipis, G., Han, E.H.: Concept indexing: a fast dimensionality reduction algorithm with applications to document retrieval and categorisation, University of Minnesota, Technical Report TR-00-0016, 2000.
12. Lang, K.: Learning to Filter netnews In Proceedings of 12th International Conference on Machine Learning, pp 331-339, 1995.
13. Lin, J: Divergence Measures Based on the Shannon Entropy, IEEE Transactions on Information Theory, 37(1), pp145-151, 1991.
14. Liu, X., Gong, Y., Xu, W., Zhu, S: Document clustering with cluster refinement and model selection capabilities. In Proceedings of SIGIR-02, 25th ACM International Conference on Research and Development in Information Retrieval, pp. 191-198,2002.
15. Pantel, P. ,Lin, D.: Document clustering with committees. In the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR),2002.
16. Pereira, F., Tishby, N., Lee L.: Distributional clustering of English words. In 30th Annual Meeting of the Association for Computational Linguistics, Columbus. Ohio, pp. 183-190, 1993.
17. Sebastiani, F.: Machine learning in automated text categorization, ACM Computer Surveys, Vol.34, No.1, March 2002, pp. 1-47, 2002.
18. Slonim, N.,Tishby N: Document Clustering using word clusters via the Information Bottleneck method. In the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2000.
19. Slonim, N.,Friedman, N., Tishby N.: Unsupervised document classification using sequential information maximization. In the 25th Annual International Conference on Research and Development in Information Retrieval (SIGIR),2002.
20. Tishby N., Pereira F., Bialek W.: The Information bottleneck method. Invited paper to The 37th annual Allerton Conference on Communication, Control, and Computing, 1999.
21. Van Rijsbergen, C. J.: Information retrieval, Butterworth-Heinemann, 1979.
22. Zamir, O. and Etzioni, O.: Web document Clustering, A feasibility demonstration in ACM SIGIR 98, pp 46-54, 1998.

Complex Linguistic Features for Text Classification: A Comprehensive Study

Alessandro Moschitti[1] and Roberto Basili[2]

[1] University of Texas at Dallas, Human Language Technology Research Institute
Richardson, TX 75083-0688, USA
alessandro.moschitti@utdallas.edu
[2] University of Rome Tor Vergata, Computer Science Department
00133 Roma (Italy),
basili@info.uniroma2.it

Abstract. Previous researches on advanced representations for document retrieval have shown that statistical *state-of-the-art* models are not improved by a variety of different linguistic representations. Phrases, word senses and syntactic relations derived by Natural Language Processing (NLP) techniques were observed ineffective to increase retrieval accuracy. For Text Categorization (TC) are available fewer and less definitive studies on the use of advanced document representations as it is a relatively new research area (compared to document retrieval).
In this paper, advanced document representations have been investigated. Extensive experimentation on representative classifiers, Rocchio and SVM, as well as a careful analysis of the literature have been carried out to study how some NLP techniques used for indexing impact TC. Cross validation over 4 different corpora in two languages allowed us to gather an *overwhelming evidence* that complex nominals, proper nouns and *word senses* are not adequate to improve TC accuracy.

1 Introduction

In the past, several attempts to design complex and effective features for document retrieval and filtering were carried out. Traditional richer representations included: document *Lemmas,* i.e. base forms of morphological categories, like nouns (e.g. *bank* from *banks*) or verbs (e.g. *work* from *worked,working*); *Phrases,* i.e. sentence fragments as word sequences; *word senses,* i.e. different meanings of content words, as defined in dictionaries.

Phrases can be divided in: (a) *simple n-grams*[1], i.e., sequences of words (e.g., *officials said*) selected by applying statistical techniques, e.g. *mutual information* or χ^2; (b) *Noun Phrases* such as Named Entities (e.g., *George Bush* or *Washington D.C.*) and other complex nominals (e.g., *satellite cable television system*); and (c) $<head, modifier_1, .., modifier_n>$ tuples in which the relations between the *head* word and modifier words are detected using syntactic parsers, e.g. [1].

[1] The term n-grams is traditionally referred to as the sequences of n characters from text but in this context they will be referred to as words sequences.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 181–196, 2004.

Typical relations (used in [2]) are *subject-verb* or *verb-object,* e.g. in *Minister announces* and *announces plans.*

The aim of phrases is to improve the precision on concept matching. For example, incorrect documents that contain the word sequence *company acquisition* are retrieved by the query *language + acquisition.* Instead, if the word sequences are replaced by the complex nominals *company acquisition* and *language acquisition,* the incorrect documents will not be retrieved since partial matches are not triggered.

Word senses can be defined in two ways: (a) by means of an explanation, like in a dictionary entry or (b) by using other words that share the same sense, like in a thesaurus, e.g. WordNet [3]. The advantage of using word senses rather than words is a more precise concept matching. For example, the verb *to raise* could refer to: (a) *agricultural texts,* when the sense is *to cultivate by growing* or (b) *economic activities* when the sense is *to raise costs.*

Phrases were experimented for the document retrieval track in TREC conferences [2,4,5,6]. The main conclusion was that the higher computational cost of the employed Natural Language Processing (NLP) algorithms prevents their application in operative IR scenario. Another important conclusion was that the experimented NLP representations can increase basic retrieval models (which use only the basic indexing model e.g., SMART) that adopt simple stems for their indexing. Instead, if advanced statistical retrieval models are used such representations do not produce any improvement [5]. In [7] was explained that pure retrieval aspects of IR, such as the statistical measures of word overlapping between queries and documents is not affected by the NLP recently developed for document indexing.

Given the above considerations, in [7] were experimented NLP resources like WordNet instead of NLP techniques. WordNet was used to define a semantic similarity function between noun pairs. As many words are polysemous, a Word Sense Disambiguation (WSD) algorithm was developed to detect the right word senses. However, positive results were obtained only after the senses were manually validated since the WSD performance, ranging between 60-70%, was not adequate to improve document retrieval. Other studies [8,9,10] report the use of word semantic information for text indexing and query expansion. The poor results obtained in [10] show that semantic information taken directly from WordNet without performing any kind of WSD is not helping IR at all. In contrast, in [11] promising results on the same task were obtained after the word senses were manually disambiguated.

In summary the high computational cost of the adopted NLP algorithms, the small improvement produced[2] and the lack of accurate WSD tools are the reasons for the failure of NLP in document retrieval. Given these outcomes, why should we try to use the same NLP techniques for TC? TC is a subtask of IR, thus, the results should be the same. However, there are different aspects of TC that require a separated study as:

[2] Due to both the NLP errors in detecting the complex structures and the use of NLP derived features as informative as the *bag-of-words.*

- In TC both set of positive and negative documents describing categories are available. This enables the application of theoretically motivated machine learning techniques that better select the document representations.
- Categories differ from queries as they are static, i.e., a predefined set of training documents stably define the target category. Feature selection techniques can, thus, be applied to select the relevant features and filtering out those produced by NLP errors. Moreover, documents contain more words than queries and this enables the adoption of statistic methods to derive their endogenous information.
- Effective WSD algorithms can be applied to documents whereas this was not the case for queries (especially for the short queries). Additionally, recent evaluation carried out in SENSEVAL [12], has shown accuracies of 70% for verbs, 75 % for adjectives and 80% for nouns. These last results, higher than those obtained in [7], make viable the adoption of semantic representation as a recent paper on the use of senses for document retrieval [13] has pointed out.
- For TC are available fewer studies that employ NLP techniques for TC as it is a relatively new research area (compared to document retrieval) and several researches, e.g. [14,15,16,17,18,19] report noticeable improvements over the *bag-of-words.*

In this paper, the impact of richer document representations on TC has been deeply investigated on four corpora in two languages by using cross validation analysis. Phrase and sense representations have been experimented on three classification systems: Rocchio [20] and the Parameterized Rocchio Classifier (PRC) described in [21,22], and SVM-light available at `http://svmlight.joachims.org/` [23,24]. Rocchio and PRC are very efficient classifiers whereas SVM is one *state-of-the-art* TC model.

We chose the above three classification systems as richer representations can be really useful only if: (a) accuracy increases with respect to the *bag-of-words* baseline for the different systems, or (b) they improve computationally efficient classifiers so that they approach the accuracy of (more complex) state-of-art models. In both cases, NLP would enhance the TC *state-of-the-art.*

Unfortunately results, in analogy with document retrieval, demonstrate that the adopted linguistic features are not able to improve TC accuracy. In the paper, Section 2 describes the NLP techniques and the features adopted in this research. In Section 3 the cross corpora/language evaluation of our document representations is reported. Explanations of why the more sophisticated features do not work as expected is here also outlined. The related work with comparative discussion is reported in Section 4, whereas final conclusions are summarized in Section 5.

2 Natural Language Feature Engineering

The linguistic features that we used to train our classifiers are POS-tag information, i.e. syntactic category of a word (nouns, verbs or adjectives), phrases and word senses.

First, we used the Brill tagger [25][3] to identify the syntactic category (POS-tag) of each word in its corresponding context. The POS information performs a first level of word disambiguation: for example for the word *book,* it decides which is the most suitable choice between categories like *Book Sales* and *Travel Agency.*

Then, we extracted two types of phrases from texts:

- Proper Nouns (PN), which identify entities participating to events described in a text. Most named entities are locations, e.g. *Rome,* persons, e.g. *George Bush* or artifacts, e.g. *Audi 80* and are tightly related to the topics.
- Complex nominals expressing domain concepts. Domain concepts are usually identified by multiwords (e.g., *bond issues* or *beach wagon*). Their detection produce a more precise set of features that can be included in the target vector space.

The above phrases increase the precision in categorization as they provide core information that the single words may not capture. Their availability is usually ensured by external resources, i.e. thesauri or glossaries. As extensive repositories are costly to be manually developed or simply missing in most domains, we used automated methods to extract both proper nouns and complex nominals from texts. The detection of proper nouns is achieved by applying a grammar that takes into a account capital letters of nouns, e.g., *International Bureau of Law.* The complex nominal extraction has been carried out using the model presented in [26]. This is based on an integration of symbolic and statistical modeling along three major steps: the detection of atomic terms *ht* (i.e. singleton words, e.g., *issue*) using IR techniques [27], the identification of admissible candidates, i.e. linguistic structures headed by *ht* (satisfying linguistically principled grammars), and the selection of the final complex nominals via a statistical filter such as the mutual information.

The phrases were extracted per category in order to exploit the specific word statistics of each domain. Two different steps were thus required: (a) a complex nominal dictionary, namely D_i, is obtained by applying the above method to training data for each single category C_i and (2) the global complex nominal set D is obtained by merging the different D_i, i.e. $D = \cup_i D_i$.

Finally, we used word senses in place of simple words as they should give a more precise sketch of what the category is concerning. For example, a document that contains the nouns *share, field* and the verb *to raise* could refer to agricultural activities, when the senses are respectively: *plowshare, agricultural field* and *to cultivate by growing.* At the same time, the document could concern economic activities when the senses of the words are: *company share, line of business* and *to raise costs.*

[3] Although newer and more complex POS-taggers have been built, its performance is quite good, i.e. ~ 95%.

As nouns can be disambiguated with higher accuracy than the other content words we decided to use sense representation only for them. We assigned the noun senses using WordNet [3]. In this dictionary words that share the same meaning *(synonyms)* are grouped in sets called *synsets*. WordNet encodes a majority of the English nouns, verbs, adjectives and adverbs (146,350 words grouped in 111,223 synsets). A word that has multiple senses belongs to several different synsets. More importantly, for each word, its senses are ordered by their frequency in the Brown corpus. This property enables the development of a simple, baseline WSD algorithm that assigns to each word its most frequent sense[4]. Since it is not known how much WSD accuracy impacts on TC accuracy, we have implemented additionally to the baseline, a WSD algorithms based on the glosses information and we used an accurate WSD algorithm, developed by the LCC, *Language Computer Corporation* (`www.languagecomputer.com`). This algorithm is an enhancement of the one that won the SENSEVAL competition [12].

The gloss-based algorithm exploits the glosses that define the meaning of each synset. For example, the gloss of the synset $\{hit, noun\}_{\#1}$ which represents the first meaning of the noun *hit* is:
(a successful stroke in an athletic contest (especially in baseball); *"he came all the way around on Williams' hit").*
Typically, the gloss of a synset contains three different parts: (1) the definition, e.g., a *successful stroke in an athletic contest;* (2) a comment (*especially in baseball*); and (3) an example "*he came all the way around on Williams' hit*". We process only the definition part by considering it as a *local context*, whereas the document where the target noun appears is considered as a *global context.* Our semantic disambiguation function selects the sense whose local context (or gloss) *best* matches the global context. The matching is performed by counting the number of nouns that are in both the gloss and the document.

3 Experiments on Linguistic Features

We subdivided our experiments in two steps: (1) the evaluation of phrases and POS information, carried out via Rocchio PRC and SVM over *Reuters3,* Ohsumed and ANSA collections and (2) the evaluation of semantic information carried out using *SVM*[5] on *Reuters-21578* and 20NewsGroups corpora.

3.1 Experimental Set-Up

We adopted the following collections:

- The *Reuters-21578* corpus, Apté split, (`http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`). It includes 12,902 documents for 90 classes with a fixed split between testing and training (3,299 vs. 9,603).

[4] In WordNet the most frequent sense is the first one.

[5] Preliminary experiments using Rocchio and PRC on word senses showed a clear lowering of performances.

- The *Reuters3* corpus [28] prepared by Y. Yang and colleagues (`http://moscow.mt.cs.cmu.edu:8081/reuters 21450/apte`). It includes 11,099 documents for 93 classes, with a split of 3,309 vs. 7,789 between testing and training.
- The ANSA collection [22], which includes 16,000 news items in Italian from the ANSA news agency. It makes reference to 8 target categories (2,000 documents each).
- The Ohsumed collection (`ftp://medir.ohsu.edu/pub/ohsumed`), including 50,216 medical abstracts. The first 20,000 documents, categorized under the 23 *MeSH diseases* categories, have been used in our experiments.
- The 20NewsGroups corpus (20NG) available at (`http://www.ai.mit.edu/people/ jrennie/20Newsgroups/`. It contains 19997 articles for 20 categories taken from the Usenet newsgroups collection. We used only the subject and the body of each message. This corpus is different from Reuters and Ohsumed because it includes a larger vocabulary and words typically have more meanings.

To better study the impact of linguistic processing on TC, we have considered as baselines two set of tokens:

- *Tokens* set which contains a larger number of features, e.g., numbers or string with special characters. This should provide the most general *bag-of-words* results as it includes all simple features.
- *Linguistic-Tokens,* i.e. only the nouns, verbs or adjectives. These tokens are selected using the POS-information. This set is useful to measure more accurately the influence of linguistic information.

Together with the token sets we have experimented the feature sets described in Section 2, according to the following distinctions:

- Proper Nouns and Complex Nominals: +CN[6] indicates that the proper nouns and other complex nominals are used as features for the classifiers.
- Token augmented with their POS tags in context (+POS), e.g., *check*/N vs. *check*/V.

+CN denotes a set obtained by adding to the target token set, the proper nouns and complex nominals extracted from the target corpus. This results in atomic features that are simple tokens or chunked multiwords sequences (PN or CN), for which POS tag is neglected. Notice that due to their unambiguous nature, the POS tag is not critical for PN and CN. +POS+CN denotes the set obtained by taking into account POS tags for lemmas, proper nouns and complex nominals.

It is worth noting that the NLP-derived features are added to the standard token sets (instead of replacing some of them), e.g. complex nominals and proper nouns are added together with their compounding words. This choice has been

[6] Proper nouns are indeed a special case of complex nominals, thus we used a single label, i.e. +CN.

Table 1. Characteristics of Corpora used in the experiments.

Corpus	Docs	Cat.	*Tokens*	Tokens	*Ling.-*	noun senses	Lang.	*test-set*
Name				+POS+CN	*Tokens*	with BL-WSD		
*Reuters*3	11,077	93	30,424	39,840	19,000	-	Eng.	30%
Ohsumed	20,000	23	42,481	46,054	-	-	Eng.	40%
ANSA	16,000	8	56,273	69,625	-	-	Ita.	30%
Reuters-21578	12,902	90	29,103	-	-	6,794	Eng.	30%
20NGs	19,997	20	97,823	-	-	13,114	Eng.	30%

made as our previous experiments showed a decrease of classifier accuracies when the compounding words were replaced with one single *phrase-feature.* This has also been noted in other researches, e.g. [29]. The resulting corpus/feature set can be observed in Table 3.1 (the reported number of senses refers to the senses generated by the baseline WSD algorithm).

The classifiers use the *ltc* weighting scheme [27] and the following parameterization: (a) Rocchio and PRC thresholds are derived from validation sets, (b) parameters, $\beta = 16$ and $\gamma = 4$, are used for Rocchio whereas PRC estimates them on validation sets (as described in [22]) and (c) the default parameters of *SVM-light* package are used for SVM.

The performances are evaluated using the Breakeven Point (BEP) and the f_1 measure for the single categories whereas the microaverage BEP (μBEP) and the microaverage f_1 measure (μf_1) are used in case of global performances of category sets [28].

3.2 Cross-Corpora/Classifier Validations of Phrases and POS-Information

In the following we show that cross validation and the adoption of the most general token set as baseline is advisable. For example if we had used the *Linguistic-Tokens* set (nouns, verbs and adjectives) for a single experiments on the standard *Reuters3 test-set,* we would have obtained the *PRC* results shown in Table 2.

We note that both POS-tags and complex nominals produce improvements when included as features. The best model is the one using all the linguistic features. It improves the *Linguistic-Tokens* model of ~ 1.5 absolute points.

However, the baseline has been evaluated on a subset of the *Tokens* set, i.e. the *Linguistic-Tokens* set; it may produce lower performance than a more general *bag-of-words.* To investigate this aspect, in the next experiments we have added the *Tokens* set to the linguistic feature sets. We expect a reduction of the

Table 2. Breakeven points of *PRC* over *Reuters3* corpus. The linguistic features are added to the *Linguistic-Tokens* set.

	Linguistic-Tokens	+CN	+CN+POS
μBEP (93 cat.)	82.15%	83.15%	83.60%

positive impact provided by NLP since the rate of tokens sensible to linguistic processing is lowered (e.g. the POS-tags of numbers are not ambiguous).

Moreover an alternative feature set could perform higher than the *bag-of-words* in a single experiment. The classifier parameters could be better suited for a particular *training/test-set* split. Note that redundant features affect the weighting scheme by changing the norma of documents and consequently the weights of other features. Thus, to obtain more general outcomes we have cross-validated our experiments on three corpora: *Reuters3,* Ohsumed and ANSA on three classifiers Rocchio, *PRC* and *SVM* using 20 random generated splits between *test-set* (30%) and *training-set* (70%). For each split we have trained the classifiers and evaluated them on the test data. The reported performances are the average and the Std. Dev. (preceded by the ± symbol) over all 20 splits.

Table 3. Rocchio, *PRC* and *SVM* performances on different feature sets of the Reuters3 corpus

	Rocchio	*PRC*				*SVM*	
	Tokens	Tokens		+*CN*	+POS+CN	Tokens	+CN
Category	*BEP*	*BEP*	f_1	f_1	f_1	f_1	
earn	95.20	95.17	95.39	95.40	95.25	98.80	98.92
acq	80.91	86.35	86.12	87.83	87.46	96.97	97.18
money-fx	73.34	77.80	77.81	79.03	79.04	87.28	87.66
grain	74.71	88.74	88.34	87.90	87.89	91.36	91.44
crude	83.44	83.33	83.37	83.54	83.47	87.16	86.81
trade	73.38	79.39	78.97	79.72	79.59	79.13	81.03
interest	65.30	74.60	74.39	75.93	76.05	82.19	80.57
ship	78.21	82.87	83.17	83.30	83.42	88.27	88.99
wheat	73.15	89.07	87.91	87.37	86.76	83.90	84.25
corn	64.82	88.01	87.54	87.87	87.32	83.57	84.43
μf_1 (93 cat.)	80.07±0.5	84.90±0.5	84.42±0.5	84.97±0.5	84.82±0.5	88.58±0.5	88.14±0.5

Table 4. Rocchio, *PRC* and *SVM* performances on different feature sets of the Ohsumed corpus

	Rocchio	*PRC*				*SVM*	
	Tokens	Tokens		+*CN*		Tokens	+*CN*
Category	*BEP*	*BEP*	f_1	f_1	*BEP*	f_1	
Pathology	37.57	50.58	48.78	49.36	51.13	52.29	52.70
Cardiovas.	71.71	77.82	77.61	77.48	77.74	81.26	81.36
Immunologic	60.38	73.92	73.57	73.51	74.03	75.25	74.63
Neoplasms	71.34	79.71	79.48	79.38	79.77	81.03	80.81
Digest.Sys.	59.24	71.49	71.50	71.28	71.46	74.11	73.23
Neonatal	41.84	49.98	50.05	52.83	52.71	48.55	51.81
μf_1 (23 cat.)	54.36 ±0.5	66.06 ±0.4	65.81±0.4	65.90±0.4	66.32±0.4	68.43±0.5	68.36±0.5

Table 5. Rocchio and *PRC* performances on different feature sets of the ANSA corpus

	Rocchio	*PRC*		
	Tokens	Tokens	+CN	+POS+CN
Category	*BEP*	f_1	f_1	f_1
News	50.35	68.99	68.58	69.30
Economics	53.22	76.03	75.21	75.39
Politics	60.19	59.58	62.48	63.43
Entertainment	75.91	77.63	76.48	76.27
Sport	67.80	80.14	79.63	79.67
μf_1 (8 cat.)	61.76±0.5	71.00±0.4	71.80±0.4	72.37±0.4

Tables 3 shows the uselessness of POS information for *Reuters3* corpus as the measures in column 5 (+CN) and 6 (+POS+CN) assume similar values. SVM was ran on simple tokens (column 7) and on complex nominals (column 8) as they have been shown to bring more selective information in *PRC*. Similar type of evaluations are reported in tables 4 and 5.

The global performances (i.e. the microaverages) in all the tables show small improvements over the *bag-of-words* approach (*Tokens* column). For example, *PRC* improves of 84.97% - 84.42% = 0.55 that is lower than 1.45 observed in Table 2. An explanation is that the cardinality of complex nominals in these experiments is rather lower than the cardinality of *Tokens*[7] resulting in a small impact on the microaverages. The *SVM* global performances are slightly penalized by the use of *NLP-derived* features. We also note that some classes are improved by the extended features, e.g. *Neonatal Disease & Abnormalities* in Ohsumed and *Politics* or *Economic Politics* in the ANSA corpus, but this should be consider as the normal *record of cases.*

3.3 Cross Validation on Word Senses

In these experiments, we compared the SVM performances over *Tokens* against the performances over the semantic feature sets. These latter were obtained by merging the *Tokens* set with the set of disambiguated senses of the training document nouns. We used 3 different methods to disambiguate senses: the baseline, i.e. by picking-up the first sense, Alg1 that uses the gloss words and the Alg2 one of the most accurate commercial algorithm.

Additionally, we performed an indicative evaluation of these WSD algorithms on 250 manually disambiguated nouns extracted from some random *Reuters-21578* documents. Our evaluation was 78.43 %, 77.12 % and 80.55 % respectively for the baseline and the algorithms 1 and 2. As expected, the baseline has an accuracy quite high since (a) in Reuters the sense of a noun is usually the first and (b) it is easier to disambiguate nouns than verb or adjective. We note that using only the glosses, for an unsupervised disambiguation, we do not obtain systems more accurate than the baseline.

[7] There is a ratio of about 15:1 between simple tokens and complex nominals.

Table 6. Performance of SVM text classifier on the *Reuters-21578* corpus.

Category	*Tokens*	BL	Alg1	Alg2
earn	97.70±0.31	97.82±0.28	97.86±0.29	97.68±0.29
acq	94.14±0.57	94.28±0.51	94.17±0.55	94.21±0.51
money-fx	84.68±2.42	84.56±2.25	84.46±2.18	84.57±1.25
grain	93.43±1.38	93.74±1.24	93.71±1.44	93.34±1.21
crude	86.77±1.65	87.49±1.50	87.06±1.52	87.91±1.95
trade	80.57±1.90	81.26±1.79	80.22±1.56	80.71±2.07
interest	75.74±2.27	76.73±2.33	76.28±2.16	78.60±2.34
ship	85.97±2.83	87.04±2.19	86.43±2.05	86.08±3.04
wheat	87.61±2.39	88.19±2.03	87.61±2.62	87.84±2.29
corn	85.73±3.79	86.36±2.86	85.24±3.06	85.88±2.99
μf_1 (90 cat.)	87.64±0.55	88.09±0.48	87.80±0.53	87.98±0.38

Reuters-21578 and 20NewsGroups have been used in these experiments. The latter was chosen as it is richer, in term of senses, than the journalistic corpora. The performances are the average and the Std. Dev. (preceded by the ± symbol) of f_1 over 20 different splits (30% test-set and 70% training) for the single categories and the μf_1 for all category corpus.

Table 6 shows the *SVM* performances for 4 document representations: *Tokens* is the usual most general *bag-of-words,* BL stands for the baseline algorithm and Alg i stands for Algorithm i. We can notice that the presence of semantic information has globally enhanced the classifier. Surprisingly, the microaverage f-score (μf_1) of the baseline WSD method is higher than those of the more complex WSD algorithms. Instead, the ranking among Alg1 and Alg2 is the expected one. In fact, Alg2, i.e. the complex model of LCC, obtains an accuracy better than Alg1, which is a simpler algorithm based on glosses. However, these are only speculative reasoning since the values of the Standard Deviations ([0.38, 0.53]) prevent a statistical assessment of our conclusions.

Table 7. SVM μf_1 performances on 20NewsGroups.

Category	*Tokens*	BL	Alg1	Alg2
μf_1 (20 cat.)	83.38±0.33	82.91±0.38	82.86±0.40	82.95±0.36

Similar results have been obtained for 20NewGroups, i.e. adding semantic information does not improve TC. Table 7 shows that when the words are richer in term of possible senses the baseline performs lower than Alg2.

To complete the study on the word senses, instead to add them to the *Token* set, we replaced all the nouns with their (disambiguated) senses. We obtained lower performances (from 1 to 3 absolute points) than the *bag-of-words.*

3.4 Why Do Phrases and Senses Not Help?

The NLP derived phrases seems to be bring more information than *bag-of-words,* nevertheless, experiments show small improvements for weak TC algorithms, i.e. Rocchio and PRC, and no improvement for theoretically motivated machine learning algorithm, e.g., SVM. We see at least two possible properties of phrases as explanations.

(Loss of coverage). Word information cannot be easily subsumed by the phrase information. As an example, suppose that (a) in our representation, *proper nouns* are used in place of their compounding words and (b) we are designing a classifier for the *Politics* category. If the representation for the proper noun *George Bush* is only the single feature `George_Bush` then every political test document containing only the word `Bush`, will not trigger the feature `George_Bush` typical of a political texts.

(Poor effectiveness). The information added by word sequences is poorer than word set. It is worth noticing that for a word sequence to index better than its word set counterpart, two conditions are necessary: (a) words in the sequence should appear not sequentially in some incorrect documents, e.g. *George* and *Bush* appear non sequentially in a sport document and (b) all the correct documents that contain one of the compounding words (e.g. *George* or *Bush*) should at the same time contain the whole sequence *(George Bush).* Only in this case, the proper noun increases precision while preserving recall. However, this scenario also implies that *George Bush* is a strong indication of *"Politics"* while words *Bush* and *George,* in isolation, are not indicators of such (political) category. Although possible, this situation is just so unlikely in text documents: many co-references usually are triggered by specifying a more common subsequence (e.g. *Bush* for *George Bush*). The same situation occurs frequently for the complex nominals, in which the head is usually used as a short referential.

The experiments on word senses show that there is not much difference between senses and words. The more plausible explanation is that the senses of a noun in documents of a category tend to be always the same. Moreover, different categories are characterized by different words rather than different senses. The consequence is that words are sufficient surrogates of exact senses (as also pointed out in [13]). This hypothesis is also supported by the accuracy of the WSD baseline algorithm, i.e. by selecting only the most frequent sense, it achieves a performance of 78.43% on *Reuters-21578.* It seems that almost 80% of the times one sense (i.e. the first) characterizes accurately the word meaning in Reuters documents.

A general view of these phenomena is that *textual representations* (i.e. tokens/words) are always very good at capturing the overall semantics of documents, at least as good as linguistically justified representations. This is shown over all the types of linguistic information experimented, i.e. POS tags, phrases and senses. If this can be seen partially as a negative outcome of these investigations, it must said that it instead pushes for a specific research line. IR methods oriented to textual representations of document semantics should be firstly investigated and they should stress the role of words as vehicles of natural language semantics (as opposed to logic systems of semantic types, like ontologies). It suggests that a word centric approach should be adopted in IR scenarios by trying also to approach more complex linguistic phenomena, (e.g. structural properties of texts or anaphorical references) in terms of word-based representations, e.g. word clusters or generalizations in lexical hierarchies[8].

4 Related Work

The previous section has shown that the adopted NLP techniques slightly improve weak TC classifier, e.g. Rocchio. When more accurate learning algorithms are used, e.g. *SV M,* such improvements are not confirmed. *Do other advanced representations help TC?* To answer the question we examined some literature work[9] that claim to have enhanced TC using features different from simple words. Hereafter, we will discuss the reasons for such successful outcomes. In [14] advanced NLP has been applied to categorize the HTML documents. The main purpose was to recognize student home pages. For this task, the simple word *student* cannot be sufficient to obtain a high accuracy since the same word can appear, frequently, in other University pages. To overcome this problem, the AutoSlog-TS, Information Extraction system [31] was applied to automatically extract syntactic patterns. For example, from the sentence *I am a student of computer science at Carnegie Mellon University,* the patterns: *I am* <->, <-> *is student, student of* <->, and *student at* <-> are generated. AutoSlog-TS was applied to documents collected from various computer science departments and the resulting patterns were used in combination with the simple words. Two different TC models were trained with the above set of features: Rainbow, i.e. a bayesian classifier [32] and RIPPER [33]. The authors reported higher precisions when the NLP-representation is used in place of the *bag-of-words.* These improvements were only obtained for recall levels lower than 20%. It is thus to be noticed that the low coverage of linguistic patterns explains why they are so useful only in low recall *measures.* Just because of this, no evidence is provided about a general and effective implication on TC accuracy.

In [15] n-grams with $1 \leq n \leq 5$, selected by using an incremental algorithm, were used. The Web pages in two Yahoo categories, *Education* and *References,* were used as target corpora. Both categories contain a sub-hierarchy of many other classes. An individual classifier was designed for each sub-category. The set of classifiers was trained with the n-grams observed in the few training documents available. Results showed that n-grams can produce an improvement of about 1% (in terms of *Precision* and *Recall*) in the *References* and about 4 % for *Educational.* This latter outcome may represent a good improvement over the *bag-of-words.* However, the experiments are reported only on 300 documents, although cross validation was carried out. Moreover, the adopted classifier (i.e. the *Bayesian* model) is not very accurate in general. Finally, the target measures

[8] These latter, obviously, in a fully extensional interpretation.

[9] We purposely neglected the literature that did not find representation useful for TC e.g. [30].

relate to a non standard TC task: many sub-categories (e.g., 349 for *Educational*) and few features.

In [34], results on the use of n-grams over the *Reuters-21578* and 20NewsGroups corpora are reported. n-grams were, as usual, added to the compounding words to extend the *bag-of-words*. The selection of features was done using simple document frequency. Ripper was trained with both n-grams and simple words. The improvement over the *bag-of-words* representation, for *Reuters-21578* was less than 1%, and this is very similar to our experimental outcomes referred to complex nominals. For 20NewsGroups no enhancement was obtained.

Other experiments of n-grams using Reuters corpus are reported in [18], where only bigrams were considered. Their selection is slightly different from the previous work since Information Gain was used in combination with the document frequency. The experimented TC models were Naive Bayes and Maximum Entropy [35] and both were fed with bigrams and words. On *Reuters-21578,* the authors present an improvement of ~2 % for both classifiers. The accuracies were 67.07% and 68.90%[10] respectively for Naive Bayes and Maximum Entropy. The above performances (obtained with the extended features) are far lower than the *state-of-the-art.* As a consequence we can say that bigrams affect the complexity of learning (more complex feature make poor methods more performant), but they stil not impact on absolute accuracy figures. The higher improvement reported for another corpus, i.e. some *Yahoo* sub-categories, cannot be assessed, as results cannot be replicated. Note in fact comparison with experiments reported in [15] are not possible, as the set of documents and *Yahoo* categories used there are quite different.

On the contrary, [16] reports bigram-based *SV M* categorization over *Reuters-21578.* This enables the comparison with (a) a state-of-art TC algorithm and (b) other literature results over the same datasets. The feature selection algorithm that was adopted is interesting. They used the n-grams over characters to weight the words and the bigrams inside categories. For example, the sequence of characters *to build* produces the following 5-grams: “to bu”, “o bui”, “buil” and “build”. The occurrences of the n-grams *inside* and *outside* categories were employed to evaluate the n-gram scores in the target category. In turn n-gram scores are used to weight the characters of a target word. These weights are applied to select the most relevant words and bigrams. The selected sets as well as the whole set of words and bigrams were compared on *Reuters-21578* fixed *test-set.* When bigrams were added, *SV M* performed 86.2% by improving about 0.6% the adopted token set. This may be important because to our knowledge it is the first improvement on SVM using phrases. However, it is worth considering that:

- Cross validation was not applied: the fact that *SV M* is improved on the Reuters fixed *test-set* only does not prove that *SV M* is generally enhanced. In fact, using cross validation we obtained (over *Tokens*) 87.64% (similar to the results found in [36] that is higher than the bigram outcome of Raskutti et al. [16]

[10] They used only the top 12 populated categories. Dumais reported for the top 10 categories a μf_1 of 92 % for SVM [36].

– If we consider that the Std. Dev., in our and other experiments [17], are in the range [0.4,0.6], the improvement is not sufficient to statistically assess the superiority of the bigrams.
– Only, the words were used, special character strings and numbers were removed. As it has been proven in Section 3.2 they strongly affect the results by improving the unigram model. Thus we hypothesize that the baseline could be even higher than the reported one (i.e. 85.6%).

On the contrary, another corpus experimented in [16], i.e., *ComputerSelect* shows higher *SVM* μBEP when bigrams are used, i.e. 6 absolute percent points. But again the *ComputerSelect* collection is not standard. This makes difficult to replicate the results.

The above literature shows that in general the extracted phrases do not affect accuracy on the Reuters corpus. This could be related to the structure and content of its documents, as it has been also pointed out in [16]. Reuters news are written by journalists to disseminate information and hence contain few and precise words that are useful for classification, e.g., *grain* and *acquisition.* On the other hand, other corpora, e.g. *Yahoo* or *ComputerSelect,* include more technical categories with words, like *software* and *system,* which are effective only in context, e.g., *network software* and *array system.*

It is worth noticing that textual representations can here be also seen as a promising direction. In [17], the Information Bottleneck (IB), i.e. a feature selection technique that cluster similar features/words, was applied. *SVM* fed with IB derived clusters was experimented on three different corpora: *Reuters-21578,* WebKB and 20NewsGroups. Only 20NewsGroups corpus showed an improvement of performances when IB method was used. This was explained as a consequence of the corpus “complexity”. Reuters and WebKB corpora seem to require fewer features to reach optimal performance. IB can thus be adopted either to reduce the problem complexity as well as to increase accuracy by using a simpler representation space. The improvement on 20NewsGroups, using the cluster representation, was ~ 3 percent points.

5 Conclusions

This paper reports the study of advanced document representation for TC. First, the tradition related to NLP techniques for extracting linguistically motivated features from document has been followed. The most widely used features for IR, i.e. POS-tag, complex nominals, proper nouns and word senses, have been extracted.

Second, several combination of the above feature sets have been extensively experimented with three classifiers Rocchio, PRC and SVM over 4 corpora in two languages. The purpose was either to improve significantly efficient, but less accurate, classifiers, such as Rocchio and PRC, or to enhance a *state-of-the-art* classifier, i.e. SVM. The results have shown that both semantic (word senses) and syntactic information (phrases and POS-tags) cannot achieve any of our purposes. The main reasons are their poor coverage and weak effectiveness.

Phrases or word senses are well substituted by simple words as a word in a category assumes always the same sense, whereas categories differ on words rather than on word senses.

However, the outcome of this careful analysis is not a negative statement on the role of complex linguistic features in TC but suggests that the elementary textual representation based on words is very effective. We emphasize the role of words, rather than some other logical system of semantic types (e.g. ontologies), as a vehicle to capture phenomena like event extraction and anaphora resolution. Expansion (i.e. the enlargement of the word set connected to a document or query) and clustering are another dimension of the same line of thought.

References

1. Collins, M.: Three generative, lexicalized models for statistical parsing. In: Proceedings of the ACL and EACL, Somerset, New Jersey (1997) 16–23
2. Strzalkowski, T., Jones, S.: NLP track at TREC-5. In: Text REtrieval Conference. (1996)
3. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press. (1998)
4. Strzalkowski, T., Carballo, J.P.: Natural language information retrieval: TREC-6 report. In: TREC. (1997)
5. Strzalkowski, T., Stein, G.C., Wise, G.B., Carballo, J.P., Tapanainen, P., Jarvinen, T., Voutilainen, A., Karlgren, J.: Natural language information retrieval: TREC-7 report. In: TREC. (1998)
6. Strzalkowski, T., Carballo, J.P., Karlgren, J., Hulth, A., Tapanainen, P., Jarvinen, T.: Natural language information retrieval: TREC-8 report. In: TREC. (1999)
7. Smeaton, A.F.: Using NLP or NLP resources for information retrieval tasks. In Strzalkowski, T., ed.: Natural language information retrieval. Kluwer Academic Publishers, Dordrecht, NL (1999) 99–111
8. Sussua, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In New York, A.P., ed.: Proceeding of CKIM 93. (1993)
9. Voorhees, E.M.: Using wordnet to disambiguate word senses for text retrieval. In: Proceedings of SIGIR 1993, PA, USA. (1993)
10. Voorhees, E.M.: Query expansion using lexical-semantic relations. In: Proceedings of SIGIR 1994. (1994)
11. Voorhees, E.M.: Using wordnet for text retrieval. In Fellbaum, C., ed.: WordNet: An Electronic Lexical Database, The MIT Press (1998) 285–303
12. Kilgarriff, A., Rosenzweig, J.: English senseval: Report and results. In: English SENSEVAL: Report and Results. In Proceedings of the 2nd International Conference on Language Resources and Evaluation, LREC, Athens, Greece. (2000)
13. Stokoe, C., Oakes, M.P., Tait, J.: Word sense disambiguation in information retrieval revisited. In: Proceedings of SIGIR03, Canada. (2003)
14. Furnkranz, J., Mitchell, T., Rilof, E.: A case study in using linguistic phrases for text categorization on the www. In: AAAI/ICML Workshop. (1998)
15. Mladenić, D., Grobelnik, M.: Word sequences as features in text-learning. In: Proceedings of ERK98, Ljubljana, SL (1998)
16. Raskutti, B., Ferrá, H., Kowalczyk, A.: Second order features for maximising text classification performance. In: Proceedings of ECML-01, 12th European Conference on Machine Learning, Springer Verlag, Heidelberg, DE (2001)

17. Bekkerman, R., El-Yaniv, R., Tishby, N., Winter, Y.: On feature distributional clustering for text categorization. In: Proceedings of the ACM SIGIR 2001, ACM Press (2001) 146–153
18. Tan, C.M., Wang, Y.F., Lee, C.D.: The use of bigrams to enhance text categorization. Information Processing & Management (2002)
19. Scott, S., Matwin, S.: Feature engineering for text classification. In: Proceedings of ICML-99, Bled, SL (1999)
20. Rocchio, J.: Relevance feedback in information retrieval. In G. Salton, editor, The SMART Retrieval System–Experiments in Automatic Document Processing, pages 313-323 Englewood Cliffs, NJ, Prentice Hall, Inc. (1971)
21. Basili, R., Moschitti, A., Pazienza, M.: NLP-driven IR: Evaluating performances over text classification task. In: Proceedings of IJCAI01, USA. (2001)
22. Moschitti, A.: A study on optimal parameter tuning for Rocchio text classifier. In Sebastiani, F., ed.: Proceedings of ECIR-03, 25th European Conference on Information Retrieval, Pisa, IT, Springer Verlag (2003)
23. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
24. Joachims, T.: T. joachims, making large-scale svm learning practical. In: Advances in Kernel Methods - Support Vector Learning. (1999)
25. Brill, E.: A simple rule-based part of speech tagger. In: Proc. of the Third Applied Natural Language Processing, Povo, Trento, Italy. (1992)
26. Basili, R., De Rossi, G., Pazienza, M.: Inducing terminology for lexical acquisition. In: Preoceeding of EMNLP 97 Conference, Providence, USA. (1997)
27. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management **24(5)** (1988) 513–523
28. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval Journal (1999)
29. Caropreso, M.F., Matwin, S., Sebastiani, F.: A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In: Idea Group Publishing, Hershey, US. (2001)
30. Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In: Proceedings of SIGIR-92, Kobenhavn, DK (1992)
31. Riloff, E.: Automatically generating extraction patterns from untagged text. In: AAAI/IAAI, Vol. 2. (1996) 1044–1049
32. Mitchell, T.: Machine Learning. McGraw Hill (1997)
33. Cohen, W.W., Singer, Y.: Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems **17** (1999) 141–173
34. Furnkranz, J.: A study using n-gram features for text categorization. Technical report oefai-tr-9830, Austrian Institute for Artificial Intelligence. (1998)
35. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In: IJCAI-99 Workshop. (1999)
36. Dumais, S.T., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: Proceedings of CIKM-98, Bethesda, US, ACM Press, New York, US (1998) 148–155

Eliminating High-Degree Biased Character Bigrams for Dimensionality Reduction in Chinese Text Categorization

Dejun Xue and Maosong Sun

National Key Laboratory of Intelligent Technology and Systems
Department of Computer Science and Technology, Tsinghua University
Beijing, China 100084
xdj00@mails.tsinghua.edu.cn; lkc-dcs@mail.tsinghua.edu.cn

Abstract. High dimensionality of feature space is a main obstacle for Text Categorization *(TC)*. In a candidate feature set consisting of Chinese character bigrams, there exist a number of bigrams which are high-degree biased according to character frequencies. Usually, these bigrams are likely to survive for their strength of discriminating documents after the process of feature selection. However, most of them are useless for document categorization because of the weakness in representing document contents. The paper firstly defines a criterion to identify the high-degree biased Chinese bigrams. Then, two schemes called σ-BR_1 and σ-BR_2 are proposed to deal with these bigrams: the former directly eliminates them from the feature set whereas the latter replaces them with the corresponding significant characters involved. Experimental results show that the high-degree biased bigrams should be eliminated from the feature set, and the σ-BR_1 scheme is quite effective for further dimensionality reduction in Chinese text categorization, after a feature selection process with a *Chi-CIG* score function.

1 Introduction

Text Categorization *(TC)* plays a key role in content-based document management tasks [1]. Based on Vector Space Model and similarity theory, a variety of inductive learning methods have been applied to *TC,* such as K-Nearest Neighbor [2], Centroid-Based Classifier [3], Rocchio Classifier [4], Support Vector Machine [5], etc. In the framework of Vector Space Model, text documents are indexed to weighted feature vectors with an indexing language, such as words or phrases [1,6,7]. *TC* is then transformed to measure the similarity of vectors. As the potential features are enormous (usually about hundreds of thousands), high dimensionality of feature space is a main obstacle for *TC*. It has been demonstrated that performing dimensionality reduction before inductive learning brings three advantages: declining problem scale, improving categorization effectiveness, and avoiding overfitting.

Feature selection and feature extraction are two ways to tackle the issue of high dimensionality [1]. Feature selection tends to select a subset of features to maximize categorization effectiveness from the candidate feature set with a numerical score function [8]. Generally, a score function is a statistical measure of feature importance for categorization, which is the pivot of the processing. By adjusting the score

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 197–208, 2004.

threshold, feature selection can usually achieve aggressive dimensionality reduction. The commonly used score functions include Term Frequency *(TF, tf),* Document Frequency *(DF, df),* Chi-square *(Chi),* Correlation Coefficient, Simplified *Chi,* Mutual Information *(MI),* Information Gain (*IG*), etc [1]. Yang et al.[9] experimented with five score functions on Reuters21578 collection, and concluded that *DF, IG,* and *Chi* are more effective than the others. Following another direction, feature extraction (also known as reparameterization) focuses on creating a lower- dimension orthogonal vector space from the original space by eliminating redundancy among features. Feature clustering [7], Principal Component Analysis (*PCA*) [10], and Latent Semantic Indexing *(LSI)* [11,12] are this kind of approaches. Although *PCA* and *LSI* are attractive theoretically, previous results of the two approaches are discouraging. Furthermore, computation complexity is another obstacle for them to be applied. Feature clustering reduces feature dimensionality by grouping similar features into a cluster, which is reasonable intuitively. While Lewis [7] reported a bad result about feature clustering, its advantages were recently shown in [13, 14].

Indexing language representing text documents not only determines the size of feature set, but also impacts categorization effectiveness. It has been found that more sophisticated indexing yields worse categorization effectiveness [1, 7]. In *TC* for English documents, for example, word is a good choice for document indexing. However, as Chinese has no explicit segmentation tokens, character N-gram indexing is widely used in *TC* for Chinese document representation. Nie et al. claimed that N-grams (in particular bigrams) perform as well as, or even better than words in Chinese information retrieval [15]. Following the suggestions, some works on Chinese *TC* with character bigram indexing reported satisfied effectiveness [13, 16, 17].

In this paper, we claim that there exist a lot of Chinese character bigrams whose components are high-degree biased for document categorization. These features are useless for document categorization because of the weakness of representing document content. The paper defines a bias criterion to identify the high-degree biased bigrams. Then, it proposes two schemes to deal with them for dimensionality reduction. One is to directly eliminate the features from feature set, and the other is to replace them with the corresponding significant characters. Experimental results indicate that eliminating the high-degree biased features is quite effective for further dimensionality reduction after feature selection with *Chi-CIG* score function.

The remainder of this paper is organized as follows. In Section 2, we introduce the previous work which is the basis of the paper. In Section 3, we discuss the biased bigrams. In Section 4, we define a bias criterion to identify the high-degree biased bigrams and propose two schemes for dimensionality reduction. Section 5 is about the experiment conditions, including data set, classifier, and evaluation. In Section 6, we analyze the experimental results. Conclusions are given in the last section.

2 Previous Work

In the paper, we devise a two-stage strategy for dimensionality reduction. At the first stage, we use a score function to select the important features, and remove a large number of non-informative features and noise features with an aggressive reduction degree. Then, on the feature set formed at the first stage, we further reduce feature dimensionality by eliminating the biased features which are the focus of the paper.

For the purpose of integrity, however, we explain the *Chi-CIG* score function (This work has been accepted by another international conference recently) first.

Chi and *IG* are widely used as feature-goodness criteria for feature selection in *TC* for their attractive effectiveness [9, 12, 18]. The *Chi* statistic uses the degree of dependence between a feature and a category to measure the importance of the feature for categorization (shown as Formula 1). If occurrence of a feature is strongly dependent on few categories of documents, then it gets large *Chi* values over the categories. Otherwise, its *Chi* values are small. Over the whole category set, the *Chi* value of a feature is defined in Formula 2. Introduced from information theory, the *IG* statistic measures the goodness of a feature for categorization by evaluating the number of bits of information determined by the presence or absence of the feature in a document. As the basic *IG* criterion is not reliable enough for the features with middle-low frequency, we proposed a Constrained *IG* criterion (shortened as *CIG*) which calculates the components of *IG* value over the categories in which the feature occurs. It is formulized in Formula 3.

$$Chi(T_k, c_j) = \frac{N\left[P_d(T_k, c_j) \times P_d(\overline{T_k}, \overline{c_j}) - P_d(T_k, \overline{c_j}) \times P_d(\overline{T_k}, c_j)\right]^2}{P_d(T_k) \times P_d(c_j) \times P_d(\overline{T_k}) \times P_d(\overline{c_j})} . \tag{1}$$

$$Chi(T_k) = \max_{j=1}^{M} \left\{ Chi(T_k, c_j) \right\} . \tag{2}$$

where N is the size of training documents, $P_d(T_k, c_j)$ the number of documents belonging to category c_j and containing feature T_k over N, $P_d(T_k)$ the number of documents containing T_k over N, $P_d(c_j)$ the number of documents involved in category c_j over N, and M the size of category set.

$$\begin{aligned} CIG(T_k) = & -\sum_{j=1}^{M} P_d(c_j) \log P_d(c_j) \\ & + P_d(T_k) \sum_{j=1}^{M} \left[P_d(c_j \mid T_k) \log P_d(c_j \mid T_k) \right] \times R(c_j, T_k) \\ & + P_d(\overline{T_k}) \sum_{j=1}^{M} \left[P_d(c_j \mid \overline{T_k}) \log P_d(c_j \mid \overline{T_k}) \right] \times R(c_j, T_k) . \end{aligned} \tag{3}$$

where $R(c_j, T_k) \in \{0,1\}$ indicates whether feature T_k appears in the documents belonging to category c_j or not, and $P(c_j \mid T_k)$ is the number of documents which contain T_k in c_j over the number of documents containing T_k in training set.

In our experiments, we observe that *Chi* measure prefers to the high-frequency features which unevenly distributes over categories, while *CIG* measure favors middle-frequency features with an uneven distribution over categories. By combining the two criteria, we get a better statistic, named *Chi-CIG,* shown as Formula 4.

$$Chi\text{-}CIG(T_k) = Chi(T_k) \times CIG(T_k) . \tag{4}$$

Xue et al. [17] stated that the best feature set for document categorization comprises the features which distribute unevenly in relevant documents and irrelevant documents and represent exactly the content of documents. A feature just occupying one of the two capabilities is not considered as a good feature for categorization. For example, a common feature with a good semantic quality is weak for document categorization because of its even distribution over categories. On the contrary, a low-frequency feature exclusively occurring in few documents is good at discriminating documents, but it is bad to represent the content of documents and it brings overfitting possibly. Feature selection with weighting score functions mainly emphasizes the capability of features for document discrimination.

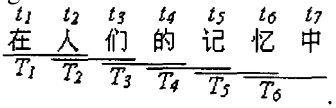
3 Biased Chinese Character Bigrams

Although the length of Chinese characters processed in computer is fixed at 2 bytes, the length of Chinese words is various. For example, in the Chinese Thesaurus (Tong Yi Ci Ci Lin) which has 52,260 Chinese words, 3874 words (7.46%) are of one-character, 33,624(64.34%) words are of two-character, and the rest 14,763 words (28.20%) consist of three or more characters [19]. Since Chinese sentences are continual strings of characters without explicit separation tokens between Chinese words, it is very difficult to identify the boundary of Chinese words in sentences. Hence, Chinese character N-gram indexing is widely used to represent Chinese text documents in automated document management tasks. In the paper, we concentrate on Chinese character bigram indexing and feature dimensionality reduction.

A number of character bigrams which are weak on representing document content have been chosen after feature selection with a sophisticated weighting score function. In out experiments (For the details of the data set, please refer to Section 5.1), for example, a Chinese sentence comprising 7 characters is:

在人们的记忆中 (in people's memory).

Its character bigram indexing is:


t_1 t_2 t_3 t_4 t_5 t_6 t_7
在 人 们 的 记 忆 中
T_1 T_2 T_3 T_4 T_5 T_6

Where, t_i is a Chinese character, and T_i a bigram feature. The statistics of the bigram features and the characters in the example are demonstrated in Table 1 and Table2 respectively. T_2 is a noun which appears in any Chinese documents, and T_1, T_3, and T_4 are common features which consist of high-frequency Chinese characters (see Table 2) and evenly distribute over category set. They are non-informative for categorization. According to the *Chi-CIG* weighting column, we can simply eliminate T_1, T_2, T_3, and T_4 by setting the weighting threshold at about 0.001 during feature selection. Of the two selected features whose weights are above the threshold, T_5 is a middle-frequency noun which has a definite meaning in the context of a document, and T_6 is a randomly formed character bigram with a low frequency. From the characters (t_6 and t_7) of T_6, people are unable to directly understand what the feature means. It indicates that T_6 is weak to represent document content. According to Xue's conclusions [17], T_6 should be worse than T_5 for categorization. However, with far less *df* and *tf*, T_6 has a larger *Chi-CIG* weight than T_5 in the example.

Table 1. *df*s, *tf*s and *Chi-CIG* Weights of the Bigram Features in the Example

Feature	*df*	*tf*	*Chi-CIG*
T_1	2,063	2,604	0.0000059
T_2	7,138	13,976	0.0001039
T_3	7,047	11,169	0.0000078
T_4	1,997	2,723	0.0000007
T_5	687	2,015	0.0016729
T_6	62	113	0.0048139

The functions of the character components in feature T_6 are biased dramatically for categorization. Table 2 shows that t_6 is a low-frequency character, and t_7 a high-frequency character. Referring to Table 1, it is observed that character t_6 dominates the *df* and *tf* values of feature T_6. In our dictionary, t_7 is a trivial polysemous one-character word. Its meaning and sense are determined after combining with other words in the context of documents. For the compound words, character t_7 is in a subsidiary position. Therefore, it appears that the document discrimination capability of feature T_6 mainly comes from character t_6. Meanwhile, the *df* and *tf* of T_6 are far less than t_6's. It indicates that the representation capability of t_6 which is a word in our dictionary as well is limited dramatically in T_6. By removing character t_7 from T_6, we create a unigram feature *T'* which leads to a better balance between the capability of document content representation and the capability of document discrimination.

Table 2. *df*s and *tf*s of the Chinese Characters in the Example

Character	*df*	*tf*
t_1	59,709	523,091
t_2	45,898	332,522
t_3	18,892	54,392
t_4	64,005	2,435,391
t_5	14,202	41,556
t_6	1,434	4,180
t_7	60,353	536,317

Although feature T_5 in our dictionary is a noun, the contributions of its character components (t_5 and t_6) for categorization are biased to some extent as well. We can see from Table 1 and Table 2, about half of the occurrences of t_6 happen at the time when t_5 and t_6 adjoin in sentences. However, character t_5, which has far larger *df* and *tf*, adjoins freely with many other characters in addition to t_6. Hence, it can be assumed that majority of T_5's contributions for categorization result from character t_6 as well. With a relaxed criterion, we remove character t_5 from feature T_5, and get another unigram feature *T"* which is the same unigram as *T'*. By combining *T'* and *T"*, the processing of dimensionality reduction is implemented. This is one scheme to reduce feature dimension by dealing with biased bigrams.

Another scheme is to keep feature T_5 unchanged with a stricter bias criterion, but to directly eliminate feature T_6 which is considered as biased from feature set. The idea comes from the observation that categorization contributions of some bigram features are melted in other features, and eliminating the reduplicated features does not affect the categorization effectiveness of original feature set. In bigram indexing, a character in the midst of a sentence appears exactly in two unique bigrams. We think that the contribution of the character for categorization can be represented completely by some of the features which include the character, and other features including the character reduplicate the contribution of the character. In the example, we think that the contribution of character t_6 for categorization has been represented in T_5. Feature T_6 whose contribution for categorization is dominated by t_6 is a reduplicated feature and can be eliminated from the feature set.

4 Eliminating High-Degree Biased Character Bigrams

The method of dimensionality reduction by using biased bigrams aims to find out the features in which their character components are high-degree biased for categorization or their contributions have been represented adequately by other features. By eliminating the high-degree biased features, a new feature set which consists of the features occupying strong capabilities of representing document content and discriminating document is set up. In order to formulize the method, we first give the definitions about biased bigrams:

A Character Belongs to a Bigram ($\in$): Given a Chinese character bigram T consisting of characters t_1 and t_2. Then, we say that character t_i ($i \in \{1, 2\}$) belongs to bigram T, shortened as $t_i \in T$.

A Bigram is σ-degree Biased (σ-B): Given a document collection D, a real value $\sigma(\geq 0)$, a bigram T, and $t_i \in T$ ($i \in \{1, 2\}$). If t_is satisfy with the following constraint:

$$\frac{max\{tf(t_1),tf(t_2)\}}{min\{tf(t_1),tf(t_2)\}} \geq \sigma ,$$

then bigram T is σ-degree biased in D, shortened as T is σ-B. Here, $tf(t_i)$ is the frequency of character t_i occurring in D, and σ is a small enough non-negative real value. If T is σ-B, we can say as well that the two characters belonging to T are σ-B. The larger is the σ, the more biased the bigram T is, and the stricter the constraint of bigram bias measure is.

Significant Character and Non-Significant Character: Given a document collection D, a σ-B bigram T, and $t_i \in T$ ($i \in \{1, 2\}$). Then, $\arg\min_{t_i}\{tf(t_i)\}$ is the significant character in T, and $\arg\max_{t_i}\{tf(t_i)\}$ is the non-significant character in T.

With the previous definitions, we devise the σ-degree Biased Reduction (σ-BR) method for Chinese character bigrams by eliminating the σ-B bigrams from feature set. It is formalized in the following steps:

(1). Implement feature selection over original feature set with the *Chi-CIG* score function and select the largest score features to form a new feature set. With the new feature set, do dimensionality reduction processing over all of the document vectors in collection (including training document vectors and test document vectors);

(2). Choose a σ value, and find out the σ-B bigrams from the feature set formed in (1) according to the constraint in σ-B definition;

(3). There are two schemes to follow at this step. The first one is to eliminate the σ-B bigrams from the training document vectors and test document vectors. Then, recount the global statistics of features over the training document vectors and lead to a learning feature set. The second scheme is to replace the σ-B bigrams with their significant characters over the training document vectors and test document vectors and keep their frequencies in the vectors unchanged. Then, sum up the frequencies over the same features within vectors, and delete the duplicate features. Re-count the global statistics of features to form a feature set for learning;

(4). With the feature set formed in (3), start the classifier learning process (Refer to Section 5.2).

If the first scheme is chosen in (3), the ***σ-BR*** method is called as σ-BR_1 for clarity in our experiments. Otherwise, if the second scheme is chosen, it is called as σ-BR_2.

5 Experiment Design

5.1 Data Set

We adopt the categorization system of Encyclopedia of China as the predefined category set which comprises 55 categories. According to the categorization system, we build up a Chinese document collection consisting of 71,674 texts with about 74 million Chinese characters. Each text is manually assigned a single category label. The number of documents in the categories is quite different, ranging from 399 (Solid Earth Physics Category) to 3,374 (Biology Category). The average number of documents in a category is 1,303. The average length of documents is 921 Chinese characters. We randomly divide the document collection into two parts: 64,533 texts for training and 7,141 texts for test in proportion of 9:1.

5.2 Classifier

In the *TC* system, we adopt centroid-based classifier which is a kind of profile-based classifier, because of its simplification and fast speed [3, 17]. During bigram indexing, the rare bigrams whose frequency is below 10 are eliminated first. A feature set which consists of 412,908 bigrams and acts as the candidate feature set for our experiments is set up. After global feature selection with *Chi-CIG* function and dimensionality reduction with ***δ–BR*** method, we build a learning feature set. Then, sum up *tf* vectors within each category to get the *tf* vector for the category, and further weight the features based on the summed *tf* vectors of categories with the well-known *tf*idf* weighting criterion to create weighted feature vectors for categories. The resulting feature vectors are considered as the centroids of the categories.

Then, a document-pivoted classifier f can be set up:

$$f = \arg\max_{j=1}^{M}\left(V_j \bullet d\right). \tag{5}$$

where V_j is the centroid vector of category c_j, and d the weighted feature vector of a free document.

5.3 Evaluation

For a category, classification effectiveness is evaluated in terms of precision (Pr) and recall (Re). For the categorization system, we adopt $\mathbf{F}_1$ measure with micro-averaging of individual Pr and Re, shown as Formula 6. As for the degree of dimensionality reduction, we use aggressivity (shortened as *Agg*) as evaluation measure [1], shown as Formula 7.

$$F_1 = \frac{2\,\mathrm{Pr}\times\mathrm{Re}}{\mathrm{Pr}+\mathrm{Re}}. \tag{6}$$

$$Agg = 1 - \frac{\textit{the number of features after dimensionality reduction}}{\textit{the number of features before dimensionality reduction}}. \tag{7}$$

6 Experimental Results and Analysis

The categorization results of adopting *Chi-CIG* and *Chi* score functions to select features are plotted in Figure 1. The figure shows that feature selection using the functions achieves sharp *Aggs* without losing considerable categorization effectiveness. As *Agg* falls in the range of [83%, 93%], the categorization effectiveness declines slightly, less 1.7% for *Chi-CIG* and 1.6% for *Chi.* Of the two functions, *Chi-CIG* outperforms *Chi* significantly in the range. As *Agg* is 83%, for example, F_1-measure of *Chi-CIG* is improved by 3.1% in comparison with *Chi.* The improvement results from a better feature set formed by *Chi-CIG* which integrates the advantages of both *Chi* and *CIG.* The feature set consists of the high-frequency features which unevenly distribute between relevant documents and irrelevant documents, and the middle-frequency features which converge in few categories.

The following experiments about σ-*BR* method for dimensionality reduction are based on the feature set selected with *Chi-CIG* score function at 83% *Agg.*

Figure 2 demonstrates the *Aggs* of σ-*BR* method with diverse σ. *Agg* varies in inverse proportion to parameter σ. As σ increases, the bias criterion becomes stricter, and the size of biased bigrams declines. In the strictest case (σ=100), there are still 6% biased features in the feature set. It indicates that after *Chi-CIG* feature selection, there exist a number of features which are weak to represent document content in feature set. To set up a lower-dimension feature set, these features should further be treated. Of the two schemes in σ-*BR* method, σ-BR_1 always gains a higher *Agg* than σ-BR_2 at a same bias criterion for their different solutions for the biased bigrams. σ-BR_1 directly eliminates the biased bigrams, and σ-BR_2 replaces the biased bigrams with the significant characters. For example, as σ=50, *Agg* of σ-BR_1 scheme is 11%, by 4% higher than the *Agg* of σ-BR_2 scheme (7%).

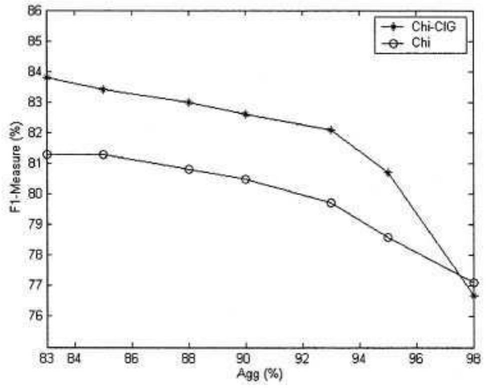


Fig. 1. Categorization Performance of *tf*idf* Weighting Criterion on Feature Sets Formed by *Chi-CIG* and *Chi* Score Functions with Diverse *Aggs*

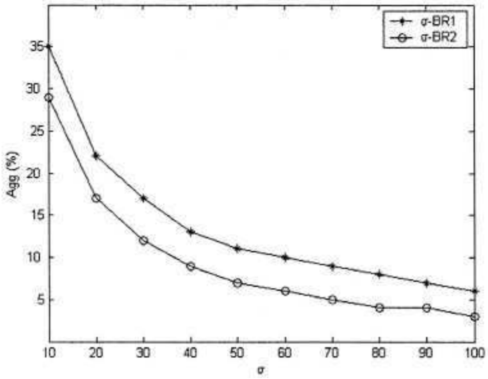


Fig. 2. *Aggs* of σ-BR_1 and σ-BR_2 Schemes for Feature Dimensionality Reduction with Diverse σ Values

Figure 3 is about the categorization effectiveness of σ-BR method. It reveals that σ-BR_1 scheme is better than, at least equals to, σ-BR_2 scheme over all σ values. Referring to the results in Figure 2, it is encouraging that in comparison with σ-BR_2 scheme, σ-BR_1 scheme increases *Aggs* of dimensionality reduction significantly without sacrificing categorization effectiveness. The observation is verified especially in the range of [40, 80] for σ values. In the range, while the *Aggs* of σ-BR_1 scheme are larger than the *Aggs* of the σ-BR_2 scheme for about 5%, σ-BR_1 scheme even improves the categorization effectiveness slightly. It is concluded that the high-degree biased bigrams should be eliminated from feature set because their functions for categorization have been contributed by other features. The conclusion coincides with the analysis in Section 4.

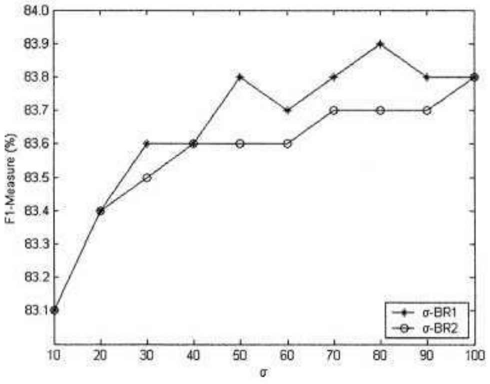


Fig. 3. Categorization Performance on Feature Sets Formed by $\boldsymbol{\sigma}$-$\boldsymbol{BR_1}$ and $\boldsymbol{\sigma}$-$\boldsymbol{BR_2}$ Schemes for Dimensionality Reduction with Diverse $\boldsymbol{\sigma}$ Values

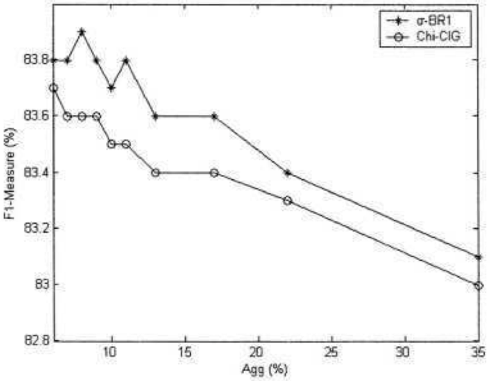


Fig. 4. Categorization Performance of *tf*idf* Weighting Criterion on Feature Sets Formed by $\boldsymbol{\sigma}$-$\boldsymbol{BR_1}$ Scheme and *Chi-CIG* Score Function for Dimensionality Reduction with Diverse *Aggs*

The comparison of categorization effectiveness of $\boldsymbol{\sigma}$-$\boldsymbol{BR_1}$ scheme and *Chi-CIG* function is illustrated in Figure 4. We can see from the curves that the categorization effectiveness of *Chi-CIG* decreases gradually with the increase of *Agg,* but the categorization effectiveness of $\boldsymbol{\sigma}$-$\boldsymbol{BR_1}$ swings at about 83.8% (baseline effectiveness) as *Agg* does not exceed 11% at which the σ value is about 50. It reveals that the high-degree biased bigram features are useless for categorization. These biased features are usually unable to identify by *Chi-CIG* measure for their powerful capability of discriminating documents, but they can be recognized by $\boldsymbol{\sigma}$-$\boldsymbol{BR_1}$ scheme. The figure shows that eliminating the high-degree biased bigram features is effective for further dimensionality reduction after feature selection with *Chi-CIG* score function.

7 Conclusions

By analyzing the characteristics of Chinese character bigrams, the paper finds that there exist a number of bigrams whose character components are high-degree biased for document categorization in feature set. After feature selection processing with sophisticated score functions, these features are usually selected for their strength of discriminating documents. Actually, however, they are not considered as outstanding features for categorization because they are weak to represent document content. The paper defines a bias criterion to identify these features. Then, two schemes called σ-*BR*$_1$ and σ-*BR*$_2$ are proposed for dimensionality reduction by dealing with the high-degree biased bigrams. σ-*BR*$_1$ scheme directly eliminates the features from feature set, and σ-*BR*$_2$ scheme replaces them with the corresponding significant characters. Experimental results on a large-scale Chinese document collection indicate that the high-degree biased features are useless for document categorization and should be eliminated from feature set. By eliminating these features, σ-*BR*$_1$ method is quite effective for further dimensionality reduction after feature selection with *Chi-CIG* score function.

The phenomenon of bias exists in English word bigrams as well. Hence, we think that the σ-*BR* method is independent of language. In the future, we plan to employ it over an English document collection to verify its correctness. Moreover, following the direction of dimensionality reduction, we are going to mine more information outside bigrams to help feature dimension reduction.

Acknowledgements. This research is supported by the National Natural Science Foundation of China under grant No. 60083005.

References

1. Fabrizio Sebastiani: Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol. 34(1). ACM Press New York (2002) 1-47
2. Yiming Yang: Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval. In Proceedings of 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1994) 11-21
3. Theeramunkong T., Lertnattee V: Improving Centroid-Based Text Classification Using Term-Distribution-Based Weighting System and Clustering. In Proceedings of International Symposium on Communications and Information Technology (2001) 33-36
4. Joachims, T: A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In Proceedings of 14th of International Conference on Machine Learning (1997) 143-151
5. Joachims T: Text Categorization with Support Vector Machines: Learnging with Many Relevant Features. In Proceedings of 10th European Conference on Machine Learning (1998) 137-142
6. Salton G., McGill M.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company, New York (1983)

7. David D. Lewis: An Evaluation of Phrasal and Clustered Representations on a Text Categorization. In Proceedings of 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1992) 37-50
8. Luis Carlos Molina, Lluis Belanche, Angela Nebot: Feature Selection Algorithms: A Survey and Experimental Evaluation. In Proceedings of 2nd IEEE International Conference on Data Mining. Maebashi City, Japan (2002) 306-313
9. Yiming Yang, Jan O. Pedersen: A Comparative Study on Feature Selection in Text Categorization. In Proceedings of 14th International Conference on Machine Learning (1997) 412-420
10. Y. H. Li, A. K. Jain: Classification of Text Document. The Computer Journal. Vol. 41, No.8, (1998)537-546
11. Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., Harshman R.: Indexing by Latent Semantic Indexing. Journal of the American Society for Information Science, Vol. 41, No.6, (1990)391-407
12. Hinrich Schutze, David A. Hull, Jan O. Pedersen: A comparison of Classifiers and Document Representations for the Routing Problem. In Proceedings of 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1995) 229-237
13. Jyh-Jong Tsay, Jing-Doo Yang: Design and Evaluation of Approaches to Automatic Chinese Text Categorization. Computational Linguistics and Chinese Language Processing, Vol. 5, No.2, August 2000, 43-58
14. Ron Bekkerman, Ran El-Yaniv, Naftali Tishby, Yoad Winter: Distributional Word Cluster vs. Words for Text Categorization. Journal of Machine Learning Research, 3(2003), 1183-1208
15. Jianyun Nie, Fuji Ren: Chinese Information Retrieval: Using Characters or Words? Information Processing and Management Vol. 35, (1999) 443-462
16. Shuigeng Zhou, Jihong Guan: Chinese Documents Classification Based on N-Grams. In Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing. Mexico City (2002) 405-414
17. Dejun Xue, Maosong Sun: A Study on Feature Weighting in Chinese Text Categorization. In Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing. Mexico City (2003) 594-604
18. Michael Oakes, Robert J. Gaizauskas, Helene Fowkes. A Method Based on the Chi-Square Test for Document Classification. In Proceedings of 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2001) 440-441
19. Shengfen Luo: Statistic-Based Two-Character Chinese Word Extraction. Master Thesis of Tsinghua University, China (2003)

Broadcast News Gisting Using Lexical Cohesion Analysis

Nicola Stokes[1], Eamonn Newman[1], Joe Carthy[1], and Alan F. Smeaton[2]

[1] Intelligent Information Retrieval Group,
Department of Computer Science, University College Dublin, Ireland.
{Nicola.Stokes,Eamonn.Newman,Joe.Carthy}@ucd.ie
[2] Centre for Digital Video Processing, Dublin City University, Ireland.
Alan.Smeaton@computing.dcu.ie

Abstract. In this paper we describe an extractive method of creating very short summaries or gists that capture the essence of a news story using a linguistic technique called lexical chaining. The recent interest in robust gisting and title generation techniques originates from a need to improve the indexing and browsing capabilities of interactive digital multimedia systems. More specifically these systems deal with streams of continuous data, like a news programme, that require further annotation before they can be presented to the user in a meaningful way. We automatically evaluate the performance of our lexical chaining-based gister with respect to four baseline extractive gisting methods on a collection of closed caption material taken from a series of news broadcasts. We also report results of a human-based evaluation of summary quality. Our results show that our novel lexical chaining approach to this problem outperforms standard extractive gisting methods.

1 Introduction

A gist is a very short summary, ranging in length from a single phrase to a sentence, that captures the essence of a piece of text in much the same way as a title or section heading in a document helps to convey the texts central message to a reader. In digital library and multimedia applications that deal with streams of continuous unmarked data tasks like text segmentation, document classification and gisting are prerequisites for the successful organisation and presentation of these data streams to users.

In this paper, we focus on creating news story gists for streams of news programmes used in the DCU Físchlár-News-Stories system [1]. In its current incarnation the Físchlár-News-Stories system segments video news streams using audio and visual analysis techniques. Like all real-world applications these techniques will at times place erroneous story boundaries in the resultant segmented video streams. In addition, since the closed caption material accompanying the video is generated live during the broadcast, a time lag exists between the discussion of the piece of news in the audio stream and the appearance of the teletext in the video stream. Consequently, segmentation errors will be present in the closed caption stream, where for example the end of one story might be merged with the beginning of the next story. Previous work in this area undertaken at the DUC summarisation workshops [2] and by other research groups has predominantly focussed on generating gists from clean data sources such as newswire [3], thus avoiding the real

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 209–222, 2004.

issue of developing techniques that can deal with the erroneous data that underlies this problem.

In Section 2 of this paper, we will discuss our approach to gisting which is based on a linguistic technique called lexical chaining, where a lexical chain is a sequence of semantically related words in a text e.g. {boat, ship, yacht, rudder, hull, bow}. The relationship between lexical cohesion analysis and lexical chaining is tackled in Section 2, while the exact details of our gisting system, the LexGister, and our novel approach to generating lexical chains in a news domain is described in Section 3. In Sections 4, 5 and 6, we describe the results of an intrinsic and automatic evaluation of our system generated gists on a collection of closed caption material taken from an Irish television news programme. We contrast these results with the performance of four baseline systems: a baseline lexical chaining approach, a *tf.idf* weighting approach, a 'lead' sentence approach, and a random extraction approach to the gisting task. Finally in Section 7, we review related title generation approaches and comment on some directions for future work.

2 Lexical Cohesion and Lexical Chaining

When reading any text it is obvious that it is not merely made up of a set of unrelated sentences, but that these sentences are in fact connected to each other in one of two ways cohesion and coherence. Lexical cohesion is the textual characteristic responsible for making the sentences of a text seem 'to hang together' [4], while coherence refers to the fact that 'there is sense in the text' [4].

Obviously coherence is a semantic relationship and needs computationally expensive processing for identification; however, cohesion is a surface relationship and is hence more accessible. Cohesion can be roughly classified into three distinct classes, *reference, conjunction* and *lexical cohesion* [5]. Conjunction is the only class, which explicitly shows the relationship between two sentences,"Mary spoke to John and he agreed with her view of the situation". Reference and lexical cohesion on the other hand indicate sentence relationships in terms of two semantically identical or related words. In the case of reference, pronouns are the most likely means of conveying referential meaning. For example, in the following sentences, "John was famished. He hadn't eaten all day", the pronoun *he* will only be understood by the reader if they refer back to the first sentence. Lexical cohesion on the other hand arises from the selection of vocabulary items and the semantic relationships between them. For example, "John went to the *supermarket* and bought some*food* for *dinner.* He also chose a nice bottle of red *wine* to accompany his *fillet steak.*" In this case cohesion is represented by the semantic relationship between the lexical items *supermarket,food, dinner, wine,* and *fillet steak.* For automatic identification of these relationships it is far easier to work with lexical cohesion than reference since more underlying implicit information is needed to discover the relationship between the pronoun in the second sentence and the word it references. Here are a number of examples taken from CNN news transcripts that illustrate the five types of lexical cohesion as defined by Halliday [5] that are present in text:

- **Repetition** occurs when a word form is repeated again in a later section of the text e.g. "In *Gaza,* though, whether the Middle East's old violent cycles continue or not, nothing will ever look quite the same once Yasir Arafat come to town. We expect him here in the *Gaza Strip* in about an hour and a half, crossing over from Egypt".

- **Repetition through synonymy** occurs when words share the same meaning but have two unique syntactical forms. "Four years ago, it passed a domestic violence act allowing *police,* not just the victims, to press charges if they believe a domestic beating took place. In the past, *officers* were frustrated, because they'd arrive on the scene of a domestic fight, there'd be a clearly battered victim and yet, frequently, there'd be no one to file charges."
- **Word association through specialisation/generalisation** occurs when a specialised/generalised form of an earlier word is used. "They've put a possible *murder weapon* in O.J. Simpson's hands; that's something that no one knew before. And it shows that he bought that *knife* more than a month or two ahead of time and you might, therefore, start the theory of premeditation and deliberation."
- **Word association through part-whole/whole-part relationships** occurs when a part-whole/whole-part relationship exists between two words e.g. *'committee'* is made up of smaller parts called *'members'*. "The Senate Finance *Committee* has just convened. *Members* had been meeting behind closed doors throughout the morning and early afternoon."
- **Statistical associations between words** occur when the nature of the association between two words cannot be defined in terms of the above relationship types. e.g. *Osama bin Laden* and *the World Trade Centre.*

One method of exploring the lexical cohesive relationships between words in a text is to build a set of lexical chains for that text. As already stated lexical chains are clusters of semantically related words, where in most cases these words are nouns. In their seminal paper on lexical chaining, Morris and Hirst [4] showed how these word clusters could be used to explore the discourse structure of a text. Since then lexical chains have be used to address a variety of NLP and IR problems including hypertext construction [6], automatic document summarization [7, 8, 9, 10], the detection of malapropisms within text [11], as an IR term weighting and indexing strategy [12, 13], and as a means of segmenting text into distinct blocks of self-contained text [14, 15]. The focus of the research in this paper is to create gists for news stories based on a lexical cohesive analysis of each story provided by a set of lexical chains.

Most lexical chaining research has involved solving NLP/IR problems in a news story domain, using an online thesaurus (in most cases WordNet [16]) to capture the lexical cohesive relationships listed above. However, there are two problems associated with this approach to chaining. Firstly, WordNet does not keep an up-to-date repository of 'everyday' proper nouns like company names and political figures. The effect of this is that these parts of speech cannot participate in the chaining process and valuable information regarding the entities in a new story is ignored. In Section 3, we describe a novel lexical chaining algorithm that addresses this problem by building noun and proper noun-based lexical chains for each news story.

The second problem associated with previous lexical chaining methods relates to the omission of statistical word associations during the chaining process, which represent a large portion of lexical cohesive relationships in text. To address this problem our chaining algorithm uses co-occurrence statistics generated from an auxiliary corpus (TDT1 corpus [17]) using the log-likelihood association metric. These additional lexical cohesive relationships (amounting to 3,566 nouns that have an average of 7 collocates each) provide our chaining algorithm with many 'intuitive' word relationships which are not supported in the WordNet taxonomy like the relationships between the following set of words, {abuse, victim, allegation, abuser}. In Section 5, we describe the results of an experiment that verifies that these enhancements, when added to a basic chaining algorithm improve the performance of our gisting system described in the following section.

3 The LexGister

In this section we present our news gister, the LexGister system. This system takes a closed caption news story and returns a one-sentence gist or headline for the story. The system consists of three components a ‘Tokeniser’, a ‘Chainer’ which creates lexical chains, and an ‘Extractor’ that uses these chains to determine which sentence best reflects the content of that story.

3.1 The Tokeniser

The objective of the chain formation process is to build a set of lexical chains that capture the cohesive structure of each news story in the data set. Before work can begin on lexical chain identification, each segment or news story is processed by a part-of-speech tagger [18]. Once the nouns in the text have been identified, morphological analysis is then performed on these nouns; all plurals are transformed into their singular state, adjectives pertaining to nouns are nominalised and all sequences of words that match grammatical structures of compound noun phrases are extracted. This idea is based on a simple heuristic proposed by Justeson and Katz [19], which involves scanning part-of-speech tagged texts for patterns of adjacent tags that commonly match proper noun phrases like ‘White House aid’, ‘PLO leader Yasir Arafat’, and WordNet noun phrases like ‘act of god’, ‘arms deal’, and ‘partner in crime’. This process also helps to improve the accuracy of the lexical chaining algorithm by removing ambiguity from the text. For example, consider the phrase ‘New York Times’ where each individual word differs in meaning to the phrase as a whole.

In general news story proper noun phrases will not be present in WordNet, since keeping an up-to-date repository of such words is a substantial and never ending problem. However, as already stated, any remaining proper nouns are still useful to the chaining process since they provide a further means of capturing lexical cohesion in the text though repetition relationships. One problem with compound proper noun phrases is that they are less likely to have exact syntactic repetitions elsewhere in the text. Hence, we introduce into our lexical chaining algorithm a fuzzy string matcher that looks first for full syntactic match *(U.S_President ⇔ U.S_President),* then partial full-word match (U.S_*President ⇔ President*_Bush) and finally a ‘constrained’ form of partial word match between the two phrases (*cave*_dwellers ⇔ *cavers*). In summary then, the Tokeniser produces tokenised text consisting of noun and proper noun phrases including information on their location in the text i.e. sentence number. This is then given as input to the next step in the gisting process, the lexical chainer.

3.2 The Lexical Chainer

The aim of the Chainer is to find relationships between tokens (nouns, proper nouns, compound nouns, nominalized adjectives) in the data set using the WordNet thesaurus and a set of statistical word associations, and to then create lexical chains from these relationships with respect to a set of chain membership rules. The chaining procedure is based on a single-pass clustering algorithm, where the first token in the input stream forms the first lexical chain and each subsequent token is then added to an existing chain if it is related to at least one other token in that chain by any lexicographical or statistical relationships.

A stronger criterion than simple semantic similarity is imposed on the addition of a phrase to a chain, where a phrase must be added to the most recently updated and strongest[1] related chain. In addition the distance between the two tokens in the text must be less than a certain maximum number of words, depending on the strength of the relationship i.e. stronger relationships have larger distance thresholds. These system parameters are important for two reasons. Firstly, these thresholds lessen the effect of spurious chains, which are weakly cohesive chains containing misidentified word associations due to the ambiguous nature of the word forms i.e. associating *gas* with *air* when *gas* refers to a *petroleum* is an example of misidentification. The creation of these sorts of chains is undesirable as they add noise to the gisting process described in the next section.

In summary then our chaining algorithm proceeds as follows: if an ‘acceptable’ relationship exists between a token and any chain member then the token is added to that chain otherwise the token will become the seed of a new chain. This process is continued until all keywords in the text have been chained. As previously stated our novel chaining algorithm differs from previous chaining attempts [6-12, 15] in two respects:

- It incorporates genre specific information in the form of statistical word associations.
- It acknowledges the importance of considering proper nouns in the chaining process when dealing with text in a news domain.

In the next section we detail how the lexical chains derived from a news story can be used to create a headline summarising the content of that story.

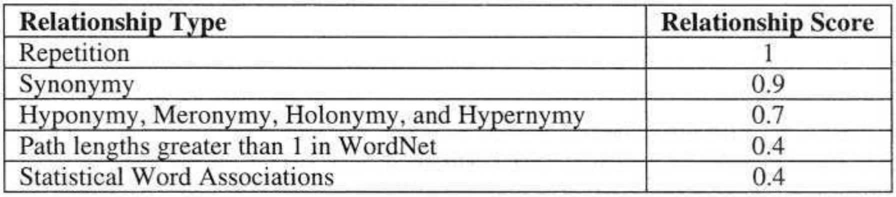
3.3 The Extractor

The final component in the LexGister system is responsible for creating a gist for each news story based on the information gleaned from the lexical chains generated in the previous phase. The first step in the extraction process is to identify the most important or highest scoring proper noun and noun chains. This step is necessary as it helps to hone in on the central themes in the text by discarding cohesively weak chains. The overall cohesive strength of a chain is measured with respect to the

[1] Relationship strength is ordered from strongest to weakest as follows: repetition, synonymy, generalisation/specialisation and whole-part/part-whole, and finally statistical word association.

strength of the relationships between the words in the chain. Table 1 shows the strength of the scores assigned to each cohesive relationship type participating in the chaining process.

Table 1. Relationship scores assigned to chain words when calculating a chain score.



Relationship Type	Relationship Score
Repetition	1
Synonymy	0.9
Hyponymy, Meronymy, Holonymy, and Hypernymy	0.7
Path lengths greater than 1 in WordNet	0.4
Statistical Word Associations	0.4

The chain weight, *score(chain),* then becomes the sum of these relationship scores, which is defined more formally as follows:

$$score(chain) = \sum_{i=1}^{n} reps(i) + rel(i,j) \qquad (1)$$

where i is the current chain word in a chain of length n, $reps(i)$ is the number of repetitions of term i in the chain and $rel(i,j)$ is the strength of the relationship between term i and the term j where j was deemed related to i during the chaining process. For example, the chain {hospital, infirmary, hospital, hospital} would be assigned a score of [*reps*(hospital) + *rel*(hospital, infirmary) + *reps*(infirmary) + *rel*(infirmary, hospital)] = 5.8, since ‘infirmary’ and ‘hospital’ are synonyms. Chain scores are not normalised, in order to preserve the importance of the length of the chain in the *score(chain)* calculation. Once all chains have been scored in this manner then the highest scoring proper noun chain and noun chain are retained for the next step in the extraction process. If the highest score is shared by more than one chain in either chain type then these chains are also retained.

Once the key noun and proper noun phrases have been identified, the next step is to score each sentence in the text based on the number of key chain words it contains:

$$score(sentence) = \sum_{i=1}^{n} score(chain)_i \qquad (2)$$

where $score(chain)_i$ is zero if word i in the current sentence of length n does not occur in one of the key chains, otherwise $score(chain)_i$ is the score assigned to the chain where i occurred.

Once all sentences have been scored and ranked, the highest ranking sentence is then extracted and used as the gist for the news article[2]. This final step in the extraction process is based on the hypothesis that the key sentence in the text will contain the most key chain words. This is analogous to saying that the key sentence should be the sentence that is most cohesively strong with respect to the rest of the text. If it happens that more than one sentence has been assigned the maximum sentence score then the sentence nearest the start of the story is chosen, since lead sentences in a news story tend to be better summaries of its content. Another

[2] At this point in the algorithm it would also be possible to generate longer-style summaries by selecting the top n ranked sentences.

consideration in the extraction phase is the occurrence of dangling anaphors in the extracted sentence e.g. references to pronoun like ‘he’ or ‘it’ that cannot be resolved within the context of the sentence. In order to address this problem we use a commonly used heuristic that states that if the gist begins with a pronoun then the previous sentence in the text is chosen as the gist. We tested the effect of this heuristic on the performance of our algorithm and found that the improvement was insignificant. We have since established that this is the case because the extraction process is biased towards choosing sentences with important proper nouns, since key proper noun chain phrases are considered. The effect of this is an overall reduction in the occurrence of dangling anaphors in the resultant gist. The remainder of the paper will discuss in more detail performance issues relating to the LexGister algorithm.

4 Evaluation Methodology

Our evaluation methodology establishes gisting performance using manual and automatic methods. The automatic evaluation is based on the same framework proposed by Witbrock and Mittal [3], where recall, precision and the F measure are used to determine the similarity of a gold standard or reference title with respect to a system generated title. In the context of this experiment these IR evaluation metrics are defined as follows:

- **Recall** (***R***) is the number of words that the reference and system titles have in common divided by the number of words in the reference title.
- **Precision** *(**P**)* is the number of words that the reference and system titles have in common divided by the number of words in the system title.
- **F measure** (***F1***) is the harmonic mean of the recall and precision metrics.

$$F1 = \frac{2(R*P)}{R+P} \tag{3}$$

In order to determine how well our lexical chaining-based gister performs the automatic part of our evaluation compares the recall, precision and F1 metrics of four baseline extractive gisting systems with the LexGister. A brief description of the techniques employed in each of these systems is now described:

- A baseline lexical chaining extraction approach **(LexGister(b))** that works in the same manner as the LexGister system except that it ignores statistical associations between words in the news story and proper nouns that do not occur in the WordNet thesaurus.
- A *tf.idf* [20] based approach **(TFIDF)** that ranks sentences in the news story with respect to the sum of their *tf.idf* weights for each word in a sentence. The *idf* statistics were generated from an auxiliary broadcast news corpus (TDT1 corpus [17]).
- A lead sentence based approach **(LEAD)** that in each case chooses the first sentence in the news story as its gist. In theory this simple method should perform well due to the pyramidal nature of news stories i.e. the most important information occurs at the start of the text followed by more detailed and less crucial information. In practice, however, due to the presence of segmentation errors in our data set, it will be shown in Section 5 that a more sophisticated approach is needed.

- A random approach **(RANDOM)** that randomly selects a sentence as an appropriate gists for each news story. This approach represents a lower bound on gisting performance for our data set.

Since the focus of our research is to design a robust technique that can gist on error prone closed caption material we manually annotated 246 RTÉ Irish broadcast news stories with titles. These titles were taken from the www.rte.ie/news website and mapped onto the corresponding closed caption version of the story, and so represent a gold standard set of titles for our news collection. The results discussed in Section 5 were generated from all 246 stories. However, due to the overhead of relying on human judges to rate gists for all of these news stories we randomly selected 100 LexGister gists for the manual part of our evaluation.

Although the F measure and the other IR based metrics give us a good indication of the quality of a gist in terms of its coverage of the main entities or events mentioned in the gold standard title, a manual evaluation involving human judges is needed to consider other important aspects of gist quality like readability and syntax. We asked six judges to rate LexGister's titles using five different quality categories ranging from 5 to 1 where 'very good = 5', 'good = 4', 'ok = 3', 'bad = 2', and 'very bad = 1'. Judges were asked to read the closed caption text for a story and then rate the LexGister headline based on its ability to capture the focus of the news story. The average score for all judges over each of the 100 randomly selected titles is then used as another evaluation metric, the results of which are discussed in Section 6. This simple scoring system was taken from another title evaluation experiment conducted by Jin and Hauptmann [21].

5 Automatic Evaluation Results

As described in Section 4 the recall, precision and F1 measures are calculated based on a comparison of the 246 generated news titles against a set of reference titles taken from the RTÉ news website. However, before the overlap between a system and reference headline for a news story is calculated both titles are stopped and stemmed using the standard InQuery stopword list [33] and the Porter stemming algorithm [34]. The decision to stop reference and system titles before comparing them is based on the observation that some title words are more important than others. For example if the reference title is 'Government still planning to introduce the proposed anti-smoking law' and the system title is 'The Vintners Association are still looking to secure a compromise' then they shares the words 'the', 'still', and 'to', then it will have successfully identified 3 out of the 9 words in the reference title, resulting in misleadingly high recall (0.33) and precision (0.3) values. Another problem with automatically comparing reference and system titles is that there may be instances of morphological variants in each title, like 'introducing' and 'introduction', that without the uses of stemming will make titles appear less similar than they actually are.

Figure 1 shows the automatic evaluation results, using the stopping and stemming method, for each of our four extractive gisting methods discussed in Section 3. For this experiment we also asked a human judge to extract the sentence that best represented the essence of each story in the test set. Hence, the F1 value 0.25 achieved by these human extracted gists represents an upper bound on gisting

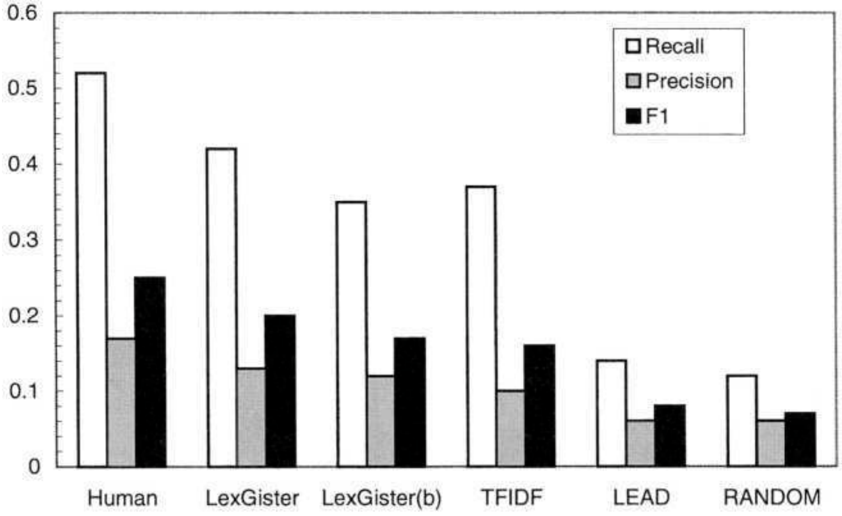


Fig. 1. Recall, Precision and F1 values measuring gisting performance for 5 distinct extractive gisting systems and a set of human extractive gists.

performance. As expected our lower bound on performance, the RANDOM system, is the worst performing system with an F1 measure of 0.07. The LEAD sentence system also performs poorly (F1 0.08), which helps to illustrate that a system that simply chooses the first sentence in this instance is not an adequate solution to the problem. A closer inspection of the collection shows that 69% of stories have segmentation errors which accounts for the low performances of the LEAD and RANDOM gisters. On the other hand, the LexGister outperforms all other systems with an F1 value of 0.20. A breakdown of this value shows a recall of 0.42, which means that on average 42% of words in a reference title are captured in the corresponding system gist generated for a news story. In contrast, the precision value for the LexGister is much lower where only 13% of words in a gist are reference title words. The precision values for the other systems show that this is a characteristic of extractive gisters since extracted sentences are on average two thirds longer than reference titles. This point is illustrated in the follow example where the recall is 100% but the precision is 50%, in both cases stopwords are ignored.

- **Gist:** "The world premier of the Veronica Guerin movie took place in Dublin's Savoy Cinema, with Cate Blanchett in the title role."
- **Reference Title:** "Premier of Veronica Guerin movie takes place in Dublin".

This example also shows that some form of sentence compression is needed if the LexGister were required to produce titles as opposed to gists, which would in turn help to increase the recall of the system. However, the high precision of the LexGister system verifies that lexical cohesion analysis is more adept at capturing the focus of a news story than a statistical-based approach using a *tf.idf* weighting scheme. Another important result from this experiment is the justification of our novel lexical chaining algorithm discuss in Section 3.2 that includes statistical word associations and proper nouns not occurring in WordNet in the chaining process. Figure 1 illustrates how the

LexGister system (F1 0.20) outperforms the baseline version, LexGister(b), using a less sophisticated lexical chaining algorithm (F1 0.17). Although our data set for this part of the experiment may be considered small in IR terms, a two-sided t-test of the null hypothesis of equal means shows that all system results are statistically significant at the 1% level, except for the difference between the RANDOM and LEAD results and the TFIDF and LexGister(b) results which are not significant.

One of the main criticisms of an automatic experiment like the one just described is that it ignores important summary attributes like readability and grammatical correctness. It also fails to recognise cases where synonymous or semantically similar words are used in a system and reference title for a news story. This is a side effect of our experimental methodology where the set of gold standard human generated titles contain many instances of words that do not occur in the original text of the news story. This makes it impossible in some cases for an extractive approach to replicate an original title. For example consider the following gists where 'Jerusalem' is replaced by 'Israel' and 'killed' is replaced by 'die': "10 killed in suicide bombing in Jerusalem" and "10 die in suicide bombing in Israel". Examples like these account for a reduction in gisting performance and illustrate how essential an intrinsic or user-oriented evaluation is for determining the 'true' quality of a gist. In the following section we describe the results of an experiment involving human judges that addresses these concerns.

6 Manual Evaluation Results

As described in Section 4, the manual evaluation of the LexGister output involves determining the quality of a gist using human judges as assessors. 100 randomly selected news stories from our closed caption data set were used for this part of the evaluation. Judges were asked to rate gists with a score ranging from 5 (a very good attempt) to 1 (a bad attempt). The average of the scores assigned by each of the six judges was then taken as the overall rating for the headlines produced by the LexGister system, where the average score was 3.56 (i.e. gists where 'ok' to 'good') with a standard deviation of 0.32 indicating strong agreement among the judges.

Since judges were asked to rate gist quality based on readability and content there were a number of situations where the gist may have captured the crux of the story but its rating was low due to problems with its fluency or readability. These problems are a side effect of dealing with error prone closed caption data that contains both segmentation errors and breaks in transmission. To estimate the impact of this problem on the rating of the titles we also asked judges to indicate if they believed that the headline encapsulated the essence of the story disregarding grammatical errors. This score was a binary decision (1 or 0), where the average judgement was that 81.33% of titles captured the central message of the story with a standard deviation of 10.52 %. This 'story essence' score suggests that LexGister headlines are in fact better than the results of the automatic evaluation suggest, since the problems resulting from the use of semantically equivalent yet syntactically different words in the system and reference titles (e.g. Jerusalem, Israel) do not apply in this case. However, reducing the number of grammatical errors in the gists is still a problem as 36% of headlines contain these sorts of errors due to 'noisy' closed caption data. An example of such an error is illustrated below where the text in italics at the beginning of the sentence has been incorrectly concatenated to the gist due to a transmission error.

> *"on tax rates relating from* Tens of thousands of commuters travelled free of charge on trains today."

It is hoped that the sentence compression strategy set out in the following section discussing Future Work will be able to remove unwanted elements of text like this from the gists. One final comment on the quality of the gists relates to the occurrence of ambiguous expressions, which occurred in 23% of system generated headlines. For example, consider the following gist which leaves the identity of 'the mountain' to the readers imagination:

> "A 34-year-old South African hotel worker collapsed and died while coming down the mountain".

To solve this problem a 'post-gisting' component would have to be developed that could replace a named entity with the longest sub-string that co-refers to it in the text [22], thus solving the ambiguous location of 'the mountain'.

Finally, a similar gisting experiment was conducted by Jin and Hauptmann [21] who found that their language modeling-based approach to title generation achieved an F1 of 0.26 and a human judgement score of 3.07 (compared with an F1 of 0.20 and a human judgement score of 3.56 for the LexGister system). Their data set consisted of 1000 documents randomly selected from the 1997 collection of broadcast news transcriptions published by Primary Source Media. All 1000 documents were used in the automatic evaluation, while 100 randomly selected documents were chosen for the manual evaluation. Although these results are not directly comparable with ours, they somewhat verify that the performance of our method is approaching the performance of other state-of-the-art broadcast news title generation systems. Also, considering that the upper bound on performance in this experiment is an F1 of 0.25 and the LexGister achieves an F1 of 0.20, this is further evidence that our approach is an adequate solution to this problem.

7 Related Research and Future Work

In this paper we have explored various extractive approaches to gisting, some other notable approaches in this area include Kraaij et al.'s [23] probabilistic approach, Alfonseca et al.'s [24] genetic algorithmic approach, and Copeck et al.'s [25] approach based on the occurrence of features that denote appropriate summary sentences. These lexical, syntactic and semantic features include the occurrence of discourse cues, the position of the sentence in the text, and the occurrence of content phrases and proper nouns. Biasing the extraction process with additional textual information like these features is a standard approach to headline generation that has proved to be highly effective in most cases [23-26].

An alternative to extractive gisting approaches is to view the title generation process as being analogous to statistical machine translation. Wittbrock and Mittal's paper on 'ultra-summarisation' [3], was one of the first attempts to generate headlines based on statistical learning methods that make use of large amounts of training data. More specifically, during title generation a news story is 'translated' into a more concise version using the Noisy Channel model. The Viterbi algorithm is then used to search for the most likely sequence of tokens in the text that would make a readable and informative headline. This is the approach adopted by Banko et al. [27], Jin and Hauptmann [21], Berger and Mittal [28] and more recently by Zajic and Dorr [29].

These researchers often state two advantages of their generative technique over an extractive one. Firstly, extractive techniques cannot deal with situations where important information may be scattered across more than one sentence. However, from an observation of our gisting results the extent of this problem may not be as pronounced as has been suggested. This is largely due to the fact that titles are so 'short and snappy' that finding the central message of the story is often sufficient and adding less important details occurring in other interesting sentences is not necessary. Also extractive techniques can work very well on gisting in a news story context as suggested by Dorr and Zajic [30] in their DUC data survey that found that the majority of headline words occur in the first sentence of a news article. The second criticism of extractive techniques related to their inability to create compact representations of a text that are smaller than a sentence. However, one of the advantages of extracting a readable and syntactically correct unit of text is that it can then be compressed using discourse analysis techniques and other linguistically rich methods. In contrast, the readability of a generated title is dependant on a 'title word ordering phrase' [3], which is based on statistical probabilities rather than any explicit consideration of grammatical correctness.

The next stage in our research is to follow the lead of current trends in title generation and use linguistically motivated heuristics to reduce a gist to a skeletal form that is grammatically and semantically correct [9, 30-32]. We have already begun working on a technique that draws on parse tree information for distinguishing important clauses in sentences using the original lexical chains generated for the news story to weight each clause. This will allow the LexGister to further hone in on which grammatical unit of the sentence is most cohesive with the rest of the news story resulting in a compact news story title. Comparing the performance of the LexGister with a generative approach to gisting is also a future goal of our research.

8 Conclusions

In this paper we have discussed our novel lexical chaining-based approach to news story gisting in the broadcast news domain. More specifically, the aim of our research is to develop a robust gisting strategy that can deal with 'noisy' closed caption material from news programmes and provide users in an interactive multimedia system with a compact headline representing the main gist of the information in a news story. We have shown the effectiveness of our technique using an intrinsic and automatic evaluation methodology. In the automatic evaluation, we compared the performance of our LexGister system to four other baseline extraction systems using recall, precision and F1 metrics to measure gist quality against a set of gold standard titles. The LexGister outperforms all systems including another lexical chaining-based gister which used a more simplistic chaining strategy. This result verifies that our novel lexical chaining approach, which incorporates both non-WordNet proper nouns and statistical word associations into the chain generation process, can greatly improve the quality of the resultant gists. The results of a user-based evaluation of gist quality also concluded that the LexGister is capable of generating informative and human readable gists for closed caption news stories.

Acknowledgements. The support of the Informatics Directorate of Enterprise Ireland is gratefully acknowledged. The authors are also grateful to the anonymous reviewers for their helpful comments.

References

1. Smeaton A.F., H. Lee, N. O'Connor, S Marlow, N. Murphy, TV News Story Segmentation, Personalisation and Recommendation. AAAI 2003 Spring Symposium on Intelligent Multimedia Knowledge Management, Stanford University 24-26 March 2003.
2. Document Understanding Conferences (DUC): www-nlpir.nist.gov/projects/duc/intro.html
3. Witbrock, M., V. Mittal, Ultra-Summarisation: A Statistical approach to generating highly condensed non-extractive summaries. In the Proceedings of the ACM-SIGIR, pp. 315-316, 1999.
4. Morris J., G. Hirst, *Lexical Cohesion by Thesaural Relations as an Indicator of the Structure of Text,* Computational Linguistics 17(1), 1991.
5. Halliday M.A.K., Spoken and Written Language. Oxford University Press, 1985.
6. Green S.J., Automatically Generating Hypertext By Comparing Semantic Similarity. University of Toronto, Technical Report number 366, October 1997.
7. Barzilay R., M. Elhadad, Using Lexical Chains for Text Summarization. In the proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, 1997.
8. Silber G.H., Kathleen F. McCoy, Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization. Computational Linguistics 28(4): 487-496, 2002.
9. Fuentes M., H. Rodriguez, L. Alonso, Mixed Approach to Headline Extraction for DUC 2003. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
10. Chali, Y., M. Kolla, N. Singh, Z. Zhang, The University of Lethbridge Text Summarizer at DUC 2003. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
11. St-Onge D., *Detecting and Correcting Malapropisms with Lexical Chains,* Dept. of Computer Science, University of Toronto, M.Sc. Thesis, 1995.
12. Stairmand M.A, A Computational Analysis of Lexical Cohesion with Applications in IR, PhD Thesis, Dept. of Language Engineering, UMIST. 1996.
13. Stokes, N., J. Carthy, First Story Detection using a Composite Document Representation, In the Proceedings of the Human Language Technology Conference, pp. 134-141, 2001.
14. Stokes N., J. Carthy, A.F. Smeaton, Segmenting Broadcast News Streams using Lexical Chains. In the Proceedings of STAIRS, pp. 145-154, 2002.
15. Okumura M., T. Honda, Word sense disambiguation and text segmentation based on lexical cohesion. In proceedings of COLING-94, pp. 755-761, 1994.
16. Miller G.A., R. Beckwith, C. Fellbaum, D. Gross, K. Miller, Five Papers on WordNet. CSL Report 43, Cognitive Science Laboratory, Princeton University, July 1990.
17. Allan J., J. Carbonell, G. Doddington, J. Yamron, Y. Yang. *Topic Detection and Tracking Pilot Study Final Report.* In the proceedings of the DARPA Broadcasting News Workshop, pp. 194-218, 1998.
18. Justeson, J. S., S.M. Katz., Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering (11): 9-27, 1995.
19. Xu J., J. Broglio, and W.B. Croft. The design and implementation of a part of speech tagger for English. Technical Report IR-52, University of Massachusetts, Amherst, Center for Intelligent Information Retrieval, 1994.

20. Salton G., M.J. McGill. Introduction to Modern Information Retrieval. New York: McGraw-Hill, 1983.
21. Jin R., A.G. Hauptmann. A new probabilistic model for title generation. In the Proceedings of the International Conference on Computational Linguistics, 2002.
22. Dimitrov, M.A light-weight approach to co-reference resolution for named entities in text, Master's Thesis, University of Sofia, 2002.
23. Kraaij, W., M. Spitters, A. Hulth. Headline extraction based on a combination of uni- and multi-document summarization techniques. In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002), 2002.
24. Alfonseca, E., P. Rodriguez. Description of the UAM system for generating very short summaries at DUC 2003. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
25. Copeck T., S. Szpakowicz. Picking phrases, picking sentences. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
26. Zhou, L., E. Hovy. Headline Summarization at ISI. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
27. Banko M., V. Mittal, M. Witbrock. Generating Headline-Style Summaries. In the Proceedings of the Association for Computational Linguistics, 2000.
28. Berger, A.L., V.O. Mittal: OCELOT: a system for summarizing Web pages. In the Proceedings of the ACM-SIGIR, pp. 144-151, 2000.
29. Zajic, D., B. Dorr. Automatic headline generation for newspaper stories. In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002), 2002.
30. Dorr, B., D. Zajic. Hedge Trimmer: A parse-and-trim approach to headline generation. In the Proceedings of the HLT/NAACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2003), 2003.
31. McKeown, K., D. Evans, A. Nenkova, R. Barzilay, V. Hatzivassiloglou, B. Schiffman, S. Blair-Goldensohn, J. Klavans, S. Sigelman. The Columbia Multi-Document Summarizer for DUC 2002. In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002), 2002.
32. Daume, H., D. Echihabi, D. Marcu, D.S. Munteanu, R. Soricut. GLEANS: A generator of logical extracts and abstracts for nice summaries. In the Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference (DUC 2002), 2002.
33. Callan J.P., W.B. Croft and S.M. Harding. *The INQUERY Retrieval System,* Database and Expert Systems Applications. In the Proceedings of the International Conference in Valencia, Spain, A.M. Tjoa and I. Ramos (ed.), Springer-Verlag, New York, 1992.
34. Porter, M.F. An algorithm for suffix stripping, *Program,* 14(3) :130-137,1980.

From Text Summarisation to Style-Specific Summarisation for Broadcast News

Heidi Christensen[1], BalaKrishna Kolluru[1], Yoshihiko Gotoh[1], and Steve Renals[2]

[1] Department of Computer Science, University of Sheffield
Sheffield S1 4DP, UK
{h.christensen, b.kolluru, y.gotoh}@dcs.shef.ac.uk

[2] The Centre for Speech Technology Research, University of Edinburgh
Edinburgh EH8 9LW, UK
srenals@inf.ed.ac.uk

Abstract. In this paper we report on a series of experiments investigating the path from text-summarisation to *style-specific* summarisation of spoken news stories. We show that the portability of traditional text summarisation features to broadcast news is dependent on the diffusiveness of the information in the broadcast news story. An analysis of two categories of news stories (containing only read speech or some spontaneous speech) demonstrates the importance of the style and the quality of the transcript, when extracting the summary-worthy information content. Further experiments indicate the advantages of doing style-specific summarisation of broadcast news.

1 Introduction

A television or radio news broadcast consists of a set of stories, containing a wide variety of content, and presented in a number of styles. A broadcast news story is often a complex composition of several elements, including both planned speech (usually read) and spontaneous speech, such as a reaction or an answer.

Printed news stories typically present the most important facts in the opening line, with subsequently related facts presented in the order of decreasing importance (the “inverted information pyramid”): indeed the opening line is often referred to as the “summary lead”. Broadcast news tend to be rather different: it is written to be heard, and the lead sentence(s) often aim to capture the interest of the viewer or listener, without summarising the main facts in the opening sentence. Furthermore, the information density within the story depends on the style: for example, the news anchor may speak information-rich sentences, compared with an interviewee. This implies that the most important information, from a summarisation viewpoint, is not distributed similarly throughout all news stories. The location of regions with a high information density may depend on the style, calling for summarisation techniques that are less rigid than those typically used for the summarisation of printed news.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 223–237, 2004.

In this paper we present some recent work in the area of broadcast news summarisation using sentence extraction techniques. The work has been aimed at investigating the path from text-summarisation to *style-specific* summarisation of spoken news stories. We have addressed three key questions:

Q1: How well do the information extraction techniques developed for text documents fare on speech? If possible, we would like to reuse textual summarisation techniques when summarising spoken language. However, speech transcripts differ from text documents in both structure and language, warranting an investigation of several issues concerning this knowledge transfer to the speech domain. We report on a series of experiments that address the performance of individual features when applied in both text and speech summarisation, as well as the effect of applying a text inspired summariser to erroneous speech recogniser transcripts (section 2). This is done by using a text corpora as well as a speech corpora, with both human ("closed-caption") and automatic speech recognition (ASR) transcripts for the broadcast TV news programmes.

Q2: To what degree is the performance of summarisers employing these text-based features dependent on the style of the broadcast news stories? There are a number of subtle differences between spontaneous and read speech [1], Stylistically, news stories with spontaneous speech tend to have the summary-worthy information distributed across the document, whereas read news stories tend to start off with a "summary lead", getting into more detail as the story progresses; adhering much more to the style of printed news stories. We have investigated the effect of the text-based features when applying a classification of news stories into two categories : stories with spontaneous elements (SPONTANEOUS) and purely read stories (READ). The analysis was carried out using the same database of spoken news stories as the first series of experiments, and effects are quantified on both "closed-caption" transcripts, low word error rate (WER) and high WER automatic transcripts (section 3).

Q3: Using the established categories (SPONTANEOUS *and* READ)*, what is the observed interaction between the summarisation technique employed and the style of news story?* The automatic summarisation techniques that we have investigated are based on sentence extraction, using novelty factor, content, and context as their respective criterion for summarisation (section 4). These automatic summarisers are compared against human generated summaries. Since we are primarily concerned with the interaction between summarisation techniques and broadcast style, in these experiments, we have used hand transcribed news broadcasts, that have been manually classified to appropriate categories, so that speech recognition errors are excluded.

2 Investigating the Portability of Text Features to the Speech Domain

For text it has been found that good extractive summarisers depend heavily on features relating to the content of the text [2] and on the structure and style of the text [3,4,5].

Content-based features are clearly vulnerable to errors introduced by a speech recognisers, and in this section we present experiments that quantify the effect of recognition errors on summarisation.

2.1 Data

Broadcast news data. For the one-sentence summarisation work we used a set of 114 ABC news broadcasts (ABC_SUM) from the TDT–2 broadcast news corpus[1], totalling 43 hours of speech. Each programme spanned 30 minutes as broadcast, reduced to around 22 minutes once advert breaks were removed, and contained on average 7–8 news stories, giving 855 stories in total. In addition to the acoustic data, both manually-generated "closed-caption" transcriptions and transcriptions from six different ASR systems (with WERs ranging from 20.5% to 32.0%), are available [7].

All ABC_SUM transcripts have been segmented at three levels: 1) sentence boundaries (hand-segmented), 2) speaker turns (produced by LIMSI [8] for TREC/SDR) and 3) story boundaries (the individual news stories were hand-segmented as part of the TREC/SDR evaluations).

For each segmented story in the ABC_SUM data, a human summariser selected one sentence as a "gold-standard", one-sentence extractive summary. These one-sentence summaries were all produced by the same human summariser, and validated in an evaluation experiment for their consistency and quality (see [9] for further details).

Two subsets of the data were used for training and developmental tests, containing 33.8 and 3.9 hours of speech respectively.

Newspaper data. We have used text data obtained from the DUC-2001[2] text summarisation evaluation. This data consists of newspaper stories originally used in the TREC–9 question answering track, totalling 144 files (132 for training, 12 for testing) from the Wall Street Journal, AP newswire, San Jose Mercury News, Financial Times, and LA Times, together with associated multi-line summaries[3]. Each document comprises a single news story topic, and the data is from the period 1987-1994. Although the speech data is from February to June 1998, the broad topics covered in the two data sets are very similar.

[1] The TDT–2 [6] corpus has been used in the NIST Topic Detection and Tracking evaluations and in the TREC–8 and TREC–9 spoken document retrieval (SDR) evaluations. The one-sentence summaries for the ABC_SUM data was developed at University of Sheffield

[2] url = http://www-nlpir.nist.gov/projects/duc/index.html

[3] Extractive summaries for this data were contributed by John Conroy (IDA) as an addition to the non-extractive summaries distributed with the original DUC-2001 data, and were derived to cover the same content as the non-extractive summaries.

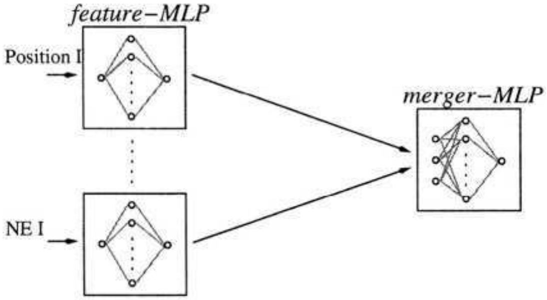


Fig. 1. Summariser architecture. All MLPs used in this work had 20 hidden units in a single hidden layer.

2.2 Summarisation Approach

The summarisation task is to automatically generate an extractive summary for a spoken or printed news story. Our approach uses a trainable, feature-based model which assigns a score to each sentence that indicates how suitable that sentence is for inclusion in a summary. When generating an *N*-line summary, the summary is comprised of the *N* highest-scoring sentences.

A set of features are extracted for each sentence. The summariser is based around a set of multi-layer perceptron (MLP) classifiers [10]: one for each feature (*feature-MLPs*) and a second level MLP (*merger-MLP*) which combines the outputs of the *feature-MLPs* (figure 1). This feature-based approach is somewhat similar to that employed by [11]; that approach discretised the features and was based on a Naive Bayes classifier. The training set for each *feature-MLP* consists of a set of single feature inputs, together with the summarisation label from the "gold-standard" (1 or 0), for each sentence. Thus each *feature-MLP* is trained to optimise summarisation for that feature alone. Given a set of trained *feature-MLPs,* a *merger-MLP* may be obtained from a training set in which each sentence is represented as the vector of *feature-MLP* outputs. This two level architecture was primarily chosen because it facilitates the analysis of the contribution of each features, by sampling the performance of the *feature-MLPs.*

We investigated a large set of candidate features, which could be divided into four categories: position of the sentence in the story, length of the sentence, similarity of the sentence to the overall document, and distribution of named entities (NEs) within the sentence. After some preliminary experiments, we settled on the set of eight features listed in table 1. The first three features can be classified as style features, and are concerned with length and position. The remaining features concern the content of the sentence. TF.IDF I and COSINE I are based on traditional information retrieval term weights comprising information about *tf (term frequency)* and *idf (inverse document frequency)* [12]. The COSINE I is the cosine similarity measure of the *tf.idf* term vector to the document term vector. The final three features all concern the NE distribution in the sentence. For the

Table 1. Description of sentence-level features. The 'start' and 'end' are relative to the boundaries of the particular news story topic. NE = named entity. Counts of NEs are per sentence. The normalised *tf.idf* features, TF.IDF I are calculated as follows: TF.IDF I $= \frac{1}{\#words} \sum_w \frac{tfidf_w}{\sqrt{\sum_{w'} tfids_{w'}}}$.

Feature	Description
POSITION I	Reciprocal position from the start.
POSITION II	Sentence position from the start.
LENGTH I	Length of sentence in words.
TF.IDF I	Mean of normalised *tf.idf* terms.
COSINE I	Cosine similarity measure of *tf.idf* terms.
NE I	Number of NEs.
NE II	Number of first occurrences of NEs.
NE III	Proportion of different NEs to number of NEs.

text data NE annotations from the DUC evaluations have been used. The speech data has been run through an automatic NE recogniser [13].

2.3 Results

We assessed the contribution of an individual feature by basing a summariser on the relevant *feature-MLP* alone. Figure 2 shows the ROC[4] curves for four of the single feature summarisers and a summariser combining the whole feature set; each operating on both text and speech data. For both text and speech the summariser based on the full feature set had the best performance characteristics. For text, the positional feature POSITION I is clearly the most informative for summarisation; for speech there is no similarly dominant feature. This is linked to the stylistic differences between print and broadcast media.

These stylistic differences are also reflected in the contribution of the last style feature, the length feature (LENGTH I). For text, the sentence length is of less importance, but for speech it contains a lot of discriminative information about whether a sentence is summary-worthy. In the speech domain, the high information regions in the stories are often from the anchor in the studio, the main reporter or the occasional expert. It is often well-formed speech with longer sentences (either read or partly scripted speech). In contrast short sentences tend to be less information-rich.

The conclusions are similar when looking at the other main group of features, the content features. In text none of these features have been able to compete with the simple, yet very effective position features. In the speech domain, the content features contribute significantly. A very noticeable difference is for the named entity based features. Their performances in the text domain are relatively poor, but again the uneven information distribution in speech means that

[4] An ROC curve depicts the relation between the false negative and true positive rates for every possible classifier output threshold.

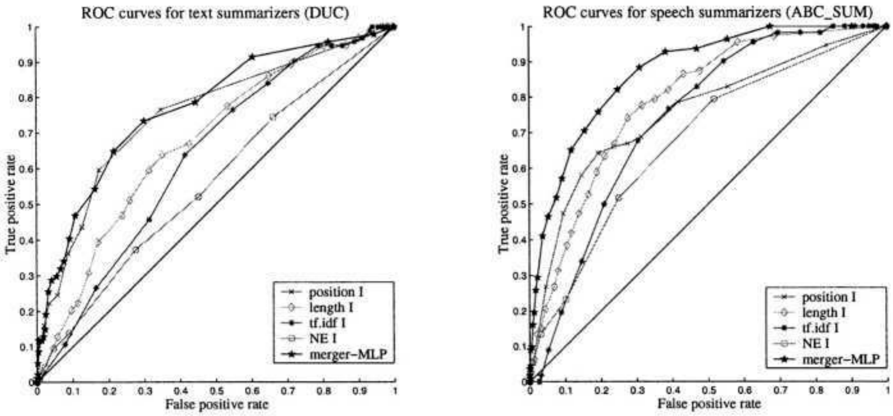


Fig. 2. Influence of the various features on the text and speech summarisers - ROC curves for the individual features and their combination to newspaper summarisation (DUC; left) and broadcast news summarisation (ABC_SUM; right).

named entities become much stronger indicators of fact filled sentences. The *tf.idf* based features tell much the same story.

A final point to note is that for text the combination of the complete eight features added only minimal improvement to the performance of the best single feature summariser - based on the simple position feature. In the speech domain, the single feature summarisers are more complementary and their combination is significantly better than any of them on their own.

Although the newspaper text and the broadcast news speech data are chosen so as to be as closely matches as possible, one crucial difference is the type of evaluation summaries: multi-line summaries for the text and one-sentence summaries for the speech. This discrepancy between data sets adds a level of complexity when drawing conclusions from these experiments. In terms of the contribution of the individual features it is likely that the apparent lack of contribution from some of the content features on the text data is partly down to the fact that when creating a multi-line summary any sentence candidate must not only be high in information relevant to the content of the story but also be a complementary match to sentences already selected in the summary.

The above experiments on broadcast news were carried out on manual, closed-caption transcriptions. Although these transcripts are not error-free (WER of 14.5%) they are still far better than transcripts from ASR systems. However, applications for automatic summarisation of spoken news stories would have to make do with transcripts output from automatic speech recognisers. Figure 3 shows the ROC curves for speech summarisers based on transcripts from six different ASR systems (produced for the TREC–8 SDR evaluation), along with the manual transcript. Each summariser was trained and tested on transcripts from the same source. The results indicate that there is relatively

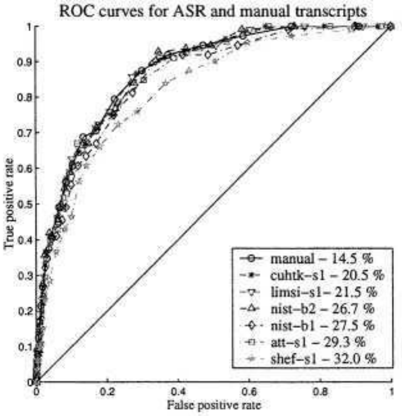


Fig. 3. The influence of various WERs on the speech data summarisers - ROC curves for summarisers corresponding to different quality ASR transcripts plus the "closed-caption" transcript [14].

little difference due to WER, although the summariser based on the recogniser with the highest WER does show some degradation in performance.

3 Information Extraction on Spontaneous and Read News Stories

The observed relative indifference to WER is similar to that observed in spoken document retrieval using this data [15], and can be explained, at least in part, by the structure of a typical broadcast news story. The most information rich regions of a broadcast news story tend to correspond to planned studio speech; whereas spontaneous speech in variable acoustic environments is less information rich, from the point of view of summarisation—and harder to recognise. Zechner *et al.* report an increase in summarisation accuracy and a decrease in WER on broadcast news summaries by taking into account the confidence score output by the ASR system when producing the summary, and thereby weighting down regions of speech with potentially high WERs [16]. Kikuchi *et al.* propose a method that in an initial stage removes sentences with low recognition accuracy and/or low significance [17].

Clearly, factors such as the structure of the news story, the WER of the transcript and the types of feature do have an effect on the summary. For ABC_SUM, the structure of the news stories varies: some have a diffuse spread of information, others are more reminiscent of newspaper stories.

Here we report on experiments that aim to quantify the effect of using traditional text features (content and style based) on automatically generated transcripts of SPONTANEOUS and READ news stories respectively. Experiments are run on three sets of transcripts from the ABC_SUM collection:

the "closed-caption" transcripts (WER = 14.5 %), transcripts from the Cambridge University HTK[5] ASR system (WER = 20.5 %) and transcripts from the Sheffield/Cambridge Abbot[6] ASR system (WER = 32.0 %).

The news stories were manually classified into SPONTANEOUS and READ stories according to the following two categories:

- **Spontaneous news:** This category includes in it all the news stories which have both planned content and spontaneous utterances made by possibly multiple subjects apart from the news-reader. Typically this category includes street interviews, question/answer based conversations and large group discussions, but may also include interviews and individual discussions.
- **Read news:** This category incorporates all the news stories whose content is pre-planned and contains no spontaneous utterance. Usually these news stories tend to be short in length compared to the other categories. Typical examples for this category are financial reports and weather reports.

These categories represent a refinement of the classification applied in [18].

3.1 Results

Figure 4 shows four plots arising from doing summarisation on SPONTANEOUS and READ news stories based on high WER, low WER and "closed-caption" transcripts. Each plot shows ROC curves from four typical *feature-MLP* summarisers as well as from the *merger-MLP* combining all eight features.

Comparing plots for the SPONTANEOUS and READ stories (left-hand column to right-hand column) shows that the different types of feature perform differently depending on the style or category of the news story. On the SPONTANEOUS stories the position feature is much less important than for the READ stories. The sentence length and the *tf.idf* based features, on the other hand, are far more important in the SPONTANEOUS stories.

Only subtle differences in summarisation accuracy arise from an increasing WER. The curves for the "closed-caption" and low WER transcripts are very similar. For the SPONTANEOUS /high WER combination the area under the ROC curves is smaller, reflecting the increased number of errors in the transcripts. A larger difference is observed for the READ /high WER stories where the length and content based features have dropped in performance, in contrast to the position feature (which is not directly dependent on the speech recogniser).

These experiments further confirm the observed link between the feature contribution and the structure of a news story, and is in line with the conclusions drawn in section 2. In our previous work, the manual classifications into SPONTANEOUS and READ were not available and instead we performed an automatic classification based on the length of the news story [9]. The results are rather similar for both cases.

[5] The cuhtk-s1 system in the 1999 TREC–8 SDR evaluation.

[6] The shef-s1 system in the 1999 TREC–8 SDR evaluation.

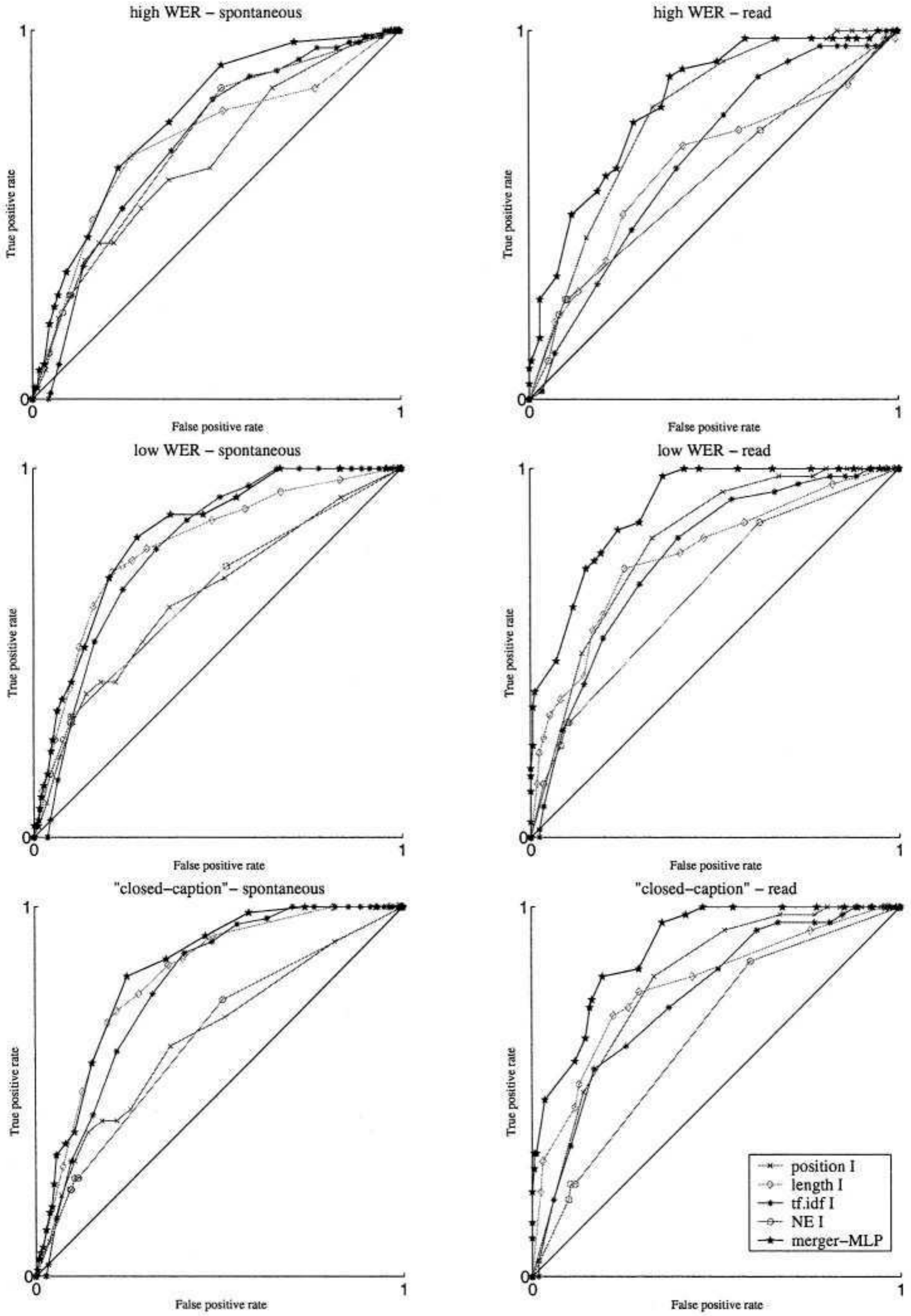


Fig. 4. The performance of summarisers based on all eight features and on four typical single feature summarisers on SPONTANEOUS and READ news stories and high WER, low WER and "closed-caption" transcripts.

4 Investigating Style-Specific Summarisation Approaches

The previous experiments have shown that the optimal choice of features when transferring text features to broadcast news is dependent on both the structure of the news story and the quality of the transcripts.

The experiments reported in this section explore the hypothesis that the observed difference in information spread in a spoken news story can be exploited, resulting in style-specific summarisation approaches.

We have used three different sentence extractive summarisers to automate this operation. The first incorporates a novelty factor to extract sentences for a summary, using an iterative technique that groups sentences which are similar to the document, but dissimilar to the partially constructed summary. The second selects the first line of the document (assumed to be a summary lead) and those sentences within the document that are similar to the first sentence. The third picks up the whole chunk of text around the sentence that is most similar to the document as a whole. For all the three summarisers we apply *tf.idf* weighting and re-arrange the selected sentences in the order of their appearance in the original document.

4.1 Data

These experiments are carried out using a portion of the hand transcripts from the Hub–4 acoustic model training data [19]. The transcripts are not case-sensitive and are devoid of any punctuation, such as sentence boundaries. For the work reported here, we manually split each segment of the transcriptions into individual news stories and marked the sentence boundaries.

For this data multi-line, "gold-standard" summaries were extracted by humans. The experiments were evaluated by a set of human judges, who scored the gold-standard summaries as well as three summaries obtained from the novelty, content and context based automatic summarisers.

4.2 Summarisation Approaches

Summariser using novelty factor. This summariser is based on the maximum marginal relevance (MMR) algorithm [2] proposed by Carbonell and Goldstein, and builds an extractive summary sentence-by-sentence, combining relevance (similarity to the document) with a novelty factor (dissimilarity to the partially constructed summary). At the k^{th} iteration, it chooses

$$s_k \equiv \hat{s} = \underset{s_i \epsilon D/E}{\operatorname{argmax}} \left\{ \lambda Sim(D, s_i) - (1 - \lambda) \max_{s_j \epsilon E} Sim(s_i, s_j) \right\} \tag{1}$$

where s_i is a sentence in the document, *D* is the document and *E* is the set of sentences already selected in the summary. *D/E* gives us the set difference: sentences not already selected. To form the summary, the selected sentences,

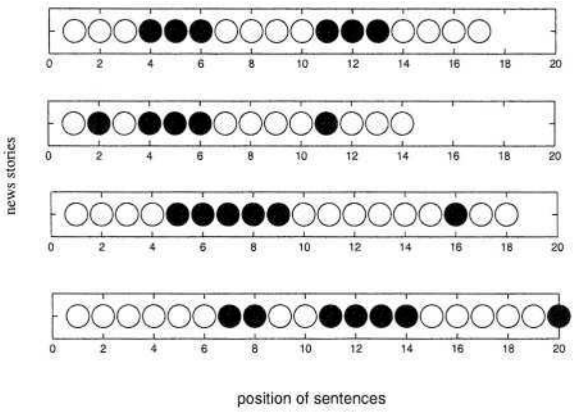


Fig. 5. Illustration showing the occurrence of sentences which are included in the summary for a given document. Each row represents a document (news story) and the sentence is represented by a circle. Each filled circle in the graph implies the sentence chosen by the human summariser to be included in the summary.

$\{s_k\}$ are re-arranged in the appearance order of the original news story. *Sim* is the cosine similarity measure.

The constant λ decides the margin for the novelty factor, thereby having a direct impact on the nature of the summary. $\lambda = 0.65$ was selected for experiments in this paper, based on some preliminary experiments on another database (BBC news transcripts).

Summariser using content. It is well-established that the first line of a textual news story is often a summary-worthy sentence, and this holds for some broadcast news stories. For example, our experiments have indicated that the first sentence is included in a human-generated extractive summary for about two-thirds of broadcast news stories. We can use this observation to design a summariser that extracts the first sentence, and treats it as a seed, extracting those other sentences that are most similar to it. The summary is a re-arrangement of $\{s_k\}$ that are selected by

$$s_k \equiv \hat{s} = \underset{s_i \epsilon D/E}{\operatorname{argmax}} \{Sim(s_1, s_i)\} \tag{2}$$

Sim is the cosine similarity measure.

Summariser using context. Another feature of extractive summarisation is that highly relevant sentences tend to occur in clusters. This is illustrated in figure 5 which shows which sentences were chosen to form part of the summary (extracted) by a human. The third summariser is based on this observation, with

Table 2. Statistics of the 22 documents (news stories) used.

categories of news stories	number of	sentences			words		
	documents	min	avg	max	min	avg	max
SPONTANEOUS	15	23	32	64	361	650	1562
READ	7	16	27	48	272	570	969

the sentence that is most similar to the whole document being chosen as a seed:

$$\hat{s} = \underset{s_i \in D}{\text{argmax}} \left\{ Sim(D, s_i) \right\}, \tag{3}$$

with the summary being formed by choosing those sentences adjacent to this seed sentence, $\hat{s}$. The summary is thus the seed sentence and its context.

4.3 Results

Each news story was manually classified into one of the two categories defined in section 3, and four summaries (three automatic, one human) were generated. Their quality was then evaluated by human judges.

We selected 22 news stories from the corpus, which were classified into the two categories. While the READ category had 7 news stories, the SPONTANEOUS news stories had 15 news stories and they varied in terms of size (Table 2). Each news story was summarised using each of the three automatic summarisers (novelty, content and context). The summarisers grouped the sentences forming a third of document or 100 words, whichever was larger.

As a benchmark, corresponding gold-standard summaries were generated by native English speakers. For the sake of uniformity of evaluation, the human summarisers were instructed to select the sentences from the document which they would ideally include in a summary, in the order of appearance.

The four summaries for each document were then rated by a set of four human judges (different from the people who summarised the documents) using a 1–10 scale, where 10 was the best. In order to obtain inter-judge agreement on the summariser, we have calculated κ [20], defined by

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{4}$$

where $P(A)$ is the proportion of the times that the l judges agree and $P(E)$ is the proportion of the times we would expect the l judges to agree by chance. Given that we are looking at l judges evaluating N document/summary pairs out of a score of a maximum of M for each category, we had to calculate the κ for each category. $P(A)$ and $P(E)$ are defined as

$$P(A) = \left[\frac{1}{Nl(l-1)} \sum_{i=1}^{N} \sum_{j=1}^{M} n_{ij}^2 \right] - \frac{1}{l-1} \tag{5}$$

Table 3. Agreement among four judges for evaluation of various summarisers.

Summariser	Human	Novelty	Content	Context
κ	0.49	0.41	0.39	0.52

where n_{ij} is the number of judges agreeing on a score of j for i^{th} summary, and

$$P(E) = \sum_{j=1}^{M} p_j^2 \tag{6}$$

where p_j is the proportion of the summaries assigned a score of j. If there is complete agreement then $\kappa = 1$ else if there is no agreement among the k raters $\kappa = 0$. The judges are said to be in moderate agreement when the κ is about 0.4 to 0.6. Table 3 shows the κ values for the four judges, indicating a moderate level of agreement.

The results of the human evaluations of the four summarisers in both the categories are shown in figure 6. The scores for each summariser in the graph are averaged over the number of stories in that category.

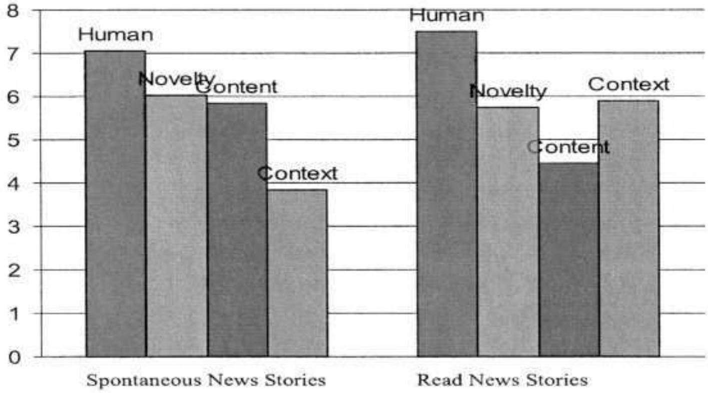


Fig. 6. The performance of the four summarisers on both categories of news stories.

The human summaries were judged to be the best for 18 out 22 stories, with the largest deviations in the respective ratings occurring in the SPONTANEOUS news story category.

The automatic summarisers using novelty and content performed similar to each other for SPONTANEOUS news stories, and they both achieved better performance than the context-based summariser. For READ news stories, the context-based summariser performs best on some stories, the novelty-based summariser is best on others; on only one READ news story was the content-based summariser the best.

The insignificance of the first line in some read news stories (see figure 5), especially in weather reports and financial reports, directly implies a drop in performance of the content summariser on READ news stories.

The context-based summariser performs better than the other two summarisers, on the READ news stories category, which has a higher density of information than the SPONTANEOUS news stories. Its performance degrades on spontaneous news stories or stories with a high degree of data sparseness. The judges pointed out that this summariser fails for this category of broadcast news stories as it fails to highlight the real issues of the document.

The novelty- and content-based summarisers tended to lack coherence, with phenomena such as unexplained subject-object references and dangling anaphora[7]. This problem is avoided by the context-based summariser, which produces more coherent summaries, but at the cost of occasional repetition.

5 Conclusions

We have investigated the portability of extractive text summarisation techniques to broadcast news. We assessed the contribution of individual features (stylistic and content-based) by investigating ROC curves for summarisers based on newspaper data and broadcast news data respectively. It was found that for text the position feature is very dominating, and features containing content information are less important. For speech however, the stylistic features and the content features were all significant.

We have shown that classical text summarisation features are largely portable to the domain of broadcast news. However, the experiments reported here also made evident that the different characteristics of a broadcast news story, such as the different information distribution and the effect of different types of transcript error, warrant more sophisticated information extraction techniques, where the organisation of summary-worthy information in the news story is more explicitly taken into consideration.

Indeed we found that different summarisers may be appropriate to different styles of news story, particularly considering whether the presentation consists of planned or spontaneous speech. The novelty-based and content-based summarisers perform well on the classes with a spontaneous element. Context-based summarisation technique is really limited to totally planned content.

We have demonstrated that various types of information extraction and summarisation approach clearly has their strength and weaknesses, which should be expoited in relation to different categories of news stories.

[7] For example, *"Another reason why. . ."* in the summary without the mention of first reason, *"and it ended tragically. . ."* without mentioning what "it" was and so on.

References

1. S. Furui, "From Read Speech Recognition to Spontaneous Speech Understanding," in *Sixth Natural Language Processing Pacific Rim Symposium,* Tokyo, Japan, 2001.
2. J. Carbonell and J. Goldstein, "The use of MMR diversity-based reranking for reordering documents and producing summaries.," in *Proceedings of ACM SIGIR,* 1998.
3. H. P. Edmundson, "New Methods in Automatic Extracting," *Journal of the ACM,* vol. 16, no. 2, pp. 264–285, 1969.
4. S. Teufel, *Argumentative Zoning: Information Extraction from Scientific Articles,* Ph.D. thesis, University of Edinburgh, 1999.
5. S. Maskey and J. Hirschberg, "Automatic Speech Summarization of Broadcast News using Structural Features," in *Proceedings of Eurospeech 2003,* Geneva, Switzerland, Sept. 2003.
6. C. Cieri, D. Graff, and M. Liberman, "The TDT-2 Text and Speech Corpus," in *Proceedings of DARPA Broadcast News Workshop,* 1999.
7. J. Garofolo, G. Auzanne, and E. Voorhees, """The TREC Spoken Document Retrieval Track: A Success Story," in *Proceedings of RIA0-2000,* Apr. 2000.
8. J.-L. Gauvain, L. Lamel, C. Barras, G. Adda and Y. de Kercardio, "The LIMSI SDR System for TREC-9," in *Proceedings of TREC-9,* 2000.
9. H. Christensen, Y. Gotoh, B. Kolluru and S. Renals, "Are extractive text summarisation techniques portable to broadcast news?," in *Proceedings of IEEE ASRU2003,* St. Thomas, US, Dec. 2003.
10. C. M. Bishop, *Neural Networks for Pattern Recognition,* Clarendon Press, Oxford, England, 1995.
11. J. Kupiec, J. O. Pedersen and F. Chen, "A Trainable Document Summarizer," in *Proceedings of ACM SIGIR,* 1995
12. C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing,* The MIT Press, 2001.
13. Y. Gotoh and S. Renals, "Information extraction from broadcast news," in *Philosophical Transactions of the Royal Society of London, series A,* vol. 358, pp. 1295–1310, Apr. 2000.
14. S. E. Johnson, P. Jourlin, K. Spärck Jones, and P. C. Woodland, "Spoken Document Retrieval for TREC-8 at Cambridge University," in *Proceedings of TREC-8,* 2000.
15. S. Renals, D. Abberley, D. Kirby, and T. Robinson, "Indexing and retrieval of broadcast news," *Speech Communication,* vol. 32, pp. 5–20, 2000.
16. K. Zechner and A. Waibel, "Minimizing Word Error Rate in Textual Summaries of Spoken Language," in *Proceedings of NAACL-ANLP-2000,* Seattle, WA, May 2000.
17. T. Kikuchi, S. Furui and C. Hori, "Automatic speech summarization based on sentence extraction and compaction," in *Proceedings of IEEE ICASSP,* Hong Kong, 2003.
18. B. Kolluru, H. Christensen, Y. Gotoh, and S. Renals, "Exploring the style-technique interaction in extractive summarization of broadcast news," in *Proceedings of IEEE ASRU2003,* St. Thomas, US, Dec. 2003.
19. "Focus Conditions for broadcast news evaluation, hub4," http://www.nist.gov/speech/tests/ bnr/ hub4_96/h4spec.htm, 1996.
20. S. Siegel and N. J. Castellan Jr, *NonParametric Statistics for the Behavioral Sciences,* McGraw-Hill International Editions, 1988.

Relevance Feedback for Cross Language Image Retrieval

Paul Clough and Mark Sanderson

Department of Information Studies, University of Sheffield, Sheffield, UK.
{p.d.clough,m.sanderson}@sheffield.ac.uk

Abstract. In this paper we show how relevance feedback can be used to improve retrieval performance for a cross language image retrieval task through query expansion. This area of CLIR is different from existing problems, but has thus far received little attention from CLIR researchers. Using the ImageCLEF test collection, we simulate user interaction with a CL image retrieval system, and in particular the situation in which a user selects one or more relevant images from the top n. Using textual captions associated with the images, relevant images are used to create a feedback model in the Lemur language model for information retrieval, and our results show that feedback is beneficial, even when only one relevant document is selected. This is particularly useful for cross language retrieval where problems during translation can result in a poor initial ranked list with few relevant in the top n. We find that the number of feedback documents and the influence of the initial query on the feedback model most affect retrieval performance.

1 Introduction

Relevance feedback is a method aimed at improving initial free-text search results by incorporating user feedback into further iterative retrieval cycles, e.g. by expanding the initial query with terms extracted from documents selected by the user (see, e.g. [1]). Typically the effect is to improve retrieval performance by either retrieving more relevant documents, or pushing existing relevant documents towards the top of a ranked list. However, this can vary greatly across different queries where the effect of additional query terms will improve some, but degrade others even though overall query expansion appears to improve retrieval (see, e.g. [2] and [3]).

The success of initial retrieval can vary due to a number of factors including: a mismatch in vocabulary between a user's search request and the document collection, the inability of a user to successfully formulate their query, and mismatches resulting from language differences between the query and document collection. The success of query expansion using relevance feedback may also vary because poor query terms are suggested as additional terms. This can be caused by factors including: the relevance of documents selected for feedback to the initial query, the selection of terms from non-relevant passages, few available

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 238–252, 2004.

documents for feedback, and irrelevant terms being selected from relevant texts (see, e.g. [4]).

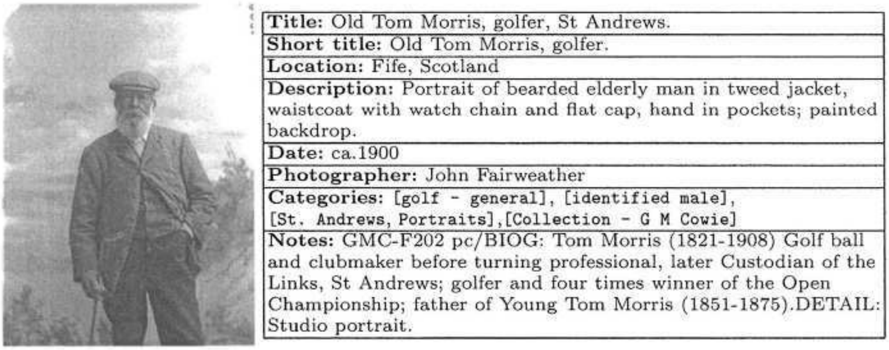
In this paper, we address the problem of matching images to user queries expressed in natural language where the images are indexed by associated textual captions. The task is cross language because the captions are expressed in one language and the queries in another, thereby requiring some kind of translation to match queries to images (e.g. query translation into the language of the document collection). Flank [5] has shown cross language image retrieval is viable on large image collections, and a program of research has been undertaken to investigate this further in the Cross Language Evaluation Forum (CLEF) [6].

The failure of query translation can vary across topics from mistranslating just one or two query terms, to not translating a term at all. Depending on the length of the query, and whether un-translated terms are important for retrieval, this can lead to unrecoverable results where the only option left to the user is to reformulate their query (or use alternative translation resources). However there are situations when enough words are translated to recover a small subset of relevant documents. In these cases relevance feedback can be used to expand the initial translated query with further terms and re-calculate existing term weights in an attempt to improve retrieval performance. Although previous work has shown that multilingual image retrieval is feasible on large collections [5], less investigation of relevance feedback in this scenario has occurred. In this paper, we pay particular attention to the situation in which initial retrieval is poor, caused by translation errors.

Most previous work in relevance feedback in CLIR has focused on pre and post-translation query expansion using blind (or pseudo) relevance feedback methods where the top n documents are assumed to be relevant (see, e.g. [8] and [9]). Although this approach involves no user interaction, it becomes ineffective when few relevant appear in the top n as non-relevant documents are also used for feedback. Past research has shown that substantial benefits can be obtained using an additional collection during pre-translation query expansion (see, e.g. [9]). However this assumes that a similar document collection can be found in the source language. Such a resource is unlikely for many image collections, such as historic photographs or commercial photographic collections, therefore we focus on what benefits post-translation expansion can achieve.

In typical CLIR scenarios, unless the user can read and understand documents in the language of the document collection, they cannot provide relevance feedback. However, image retrieval provides a unique opportunity for exploiting relevance feedback in CLIR because for many search requests, users are able to judge relevance based on the image itself without the image caption which makes relevance judgments *language independent.* In addition to this, users are able to judge the relevance of images much faster than for document retrieval, thereby allowing relevant documents in low rank positions to be used in the feedback process, which otherwise might not be examined if documents to retrieve were cross language texts. As Oard [10] comments: “an image search engine based on cross-language free text retrieval would be an excellent early application for cross-language retrieval because no translation software would be needed on the user's machine."

In these experiments we show that relevance feedback can help improve initial retrieval performance across a variety of languages, especially when translation quality is poor. We use relevance judgments from the ImageCLEF test collection to simulate the best possible user interaction. The paper is structured as follows: in section 2 we describe the evaluation framework including the language model IR system, our method of evaluating relevance feedback, the measure of retrieval effectiveness used and the translation resource employed. In section 3, we present results for initial retrieval, the variation of retrieval performance with changes in parameter values, and the effect of relevance feedback. Finally in section 4 we present our conclusions and ideas for future work.



Title: Old Tom Morris, golfer, St Andrews.

Short title: Old Tom Morris, golfer.

Location: Fife, Scotland

Description: Portrait of bearded elderly man in tweed jacket, waistcoat with watch chain and flat cap, hand in pockets; painted backdrop.

Date: ca.1900

Photographer: John Fairweather

Categories: [golf - general], [identified male], [St. Andrews, Portraits],[Collection - G M Cowie]

Notes: GMC-F202 pc/BIOG: Tom Morris (1821-1908) Golf ball and clubmaker before turning professional, later Custodian of the Links, St Andrews; golfer and four times winner of the Open Championship; father of Young Tom Morris (1851-1875).DETAIL: Studio portrait.

Fig. 1. An example image and caption from the ImageCLEF collection

2 Experimental Setup

2.1 The ImageCLEF Test Collection

The ImageCLEF test collection consists of a document collection, a set of user needs expressed in both natural language and with an exemplar image, and for each user need a set of relevance judgments [6]. The document collection consists of 28,133 images from the St Andrews Library photographic collection and all images have an accompanying textual description consisting of 8 distinct fields (see, e.g. Fig. 1). These fields can be used individually or collectively to facilitate image retrieval. The 28,133 captions consist of 44,085 terms and 1,348,474 word occurrences; the maximum caption length is 316 words, but on average 48 words in length. All captions are written in British English, although the language also contains colloquial expressions. Approximately 81% of captions contain text in all fields, the rest generally without the description field. In most cases the image description is a grammatical sentence of around 15 words. The majority of images (82%) are in black and white, although colour images are also present in the collection.

The test collection consists of 50 queries (topics) which are designed to provide a range of user requests to a cross language image retrieval system. Each topic contains a few keywords describing the user need (e.g. metal railway bridges) which have been translated into French, German, Italian, Dutch, Spanish and Chinese. The test collection also consists of a set of relevance judgments for each topic based primarily on the image, but also assisted by the image caption. Topics and relevance judgments are provided for an ad hoc retrieval task which is this: given a multilingual statement describing a user need, find as many relevant images as possible from the document collection. This retrieval task simulates when a user is able to express their need in natural language, but requires a visual document to fulfill their search request.

2.2 The Lemur Retrieval System

In the Lemur implementation of language modeling for IR, documents and queries are viewed as observations from generative[1] unigram language models (see, e.g. [11] for more information). Queries and documents are represented as estimated language models with word probabilities derived from the documents, queries and the collection as a whole. The estimated query and document language models ($\hat{\theta}_Q$ and $\hat{\theta}_D$ respectively) are compared and ranked using the KL-divergence measure, an approach which can be likened to the vector-space model of retrieval where queries and documents are represented by vectors rather than language models.

The estimated document language model $\hat{\theta}_D$ is computed from the smoothed probability of a query word seen in the document, the model smoothed by using the collection language model to estimate word probabilities when a query word is not seen in the document (see [11] for details of smoothing methods in Lemur). Without feedback, the probability of picking a word from the estimated query model $\hat{\theta}_Q$is computed using the maximum likelihood estimator based entirely on the query text. However the shorter the query, the more unstable and inaccurate this estimate will be; therefore a common method of improving this estimate is to expand the query model using relevance feedback.

By exploiting documents the user has judged as relevant (or using the top n documents from an initial search), retrieval performance is often improved because it helps to supplement the initial user query with collection-specific words obtained from a feedback model. In Lemur, the process of feedback is to update the query language model with extra evidence contained in the feedback documents, rather than simply adding additional terms to the initial query. Given an estimated feedback model $\hat{\theta}_F$ based on a set of feedback documents $F = (d_1, d_2, .., d_n)$ and the original query model $\hat{\theta}_Q$, the updated query model $\hat{\theta}_{Q'}$ is computed from interpolating the two models:

$$\hat{\theta}_{Q'} = (1 - \alpha)\hat{\theta}_Q + \alpha\hat{\theta}_F \tag{1}$$

[1] Generative in the sense that a query or document is generated by picking words from the query or document language model.

The *feedback coefficient* α controls the influence of the feedback model. If $\alpha = 0$ estimates are based entirely on the initial query; whereas if $\alpha = 1$ estimates are based entirely on the set of feedback documents. Allowing the impact of the feedback model to vary can be used to investigate the effects of including the initial query in the updated query model on retrieval performance. This is particularly relevant to cross language evaluation where the quality of the initial query is, amongst other factors, dependent on the quality of the translation resource (see [12] for details regarding estimation of the feedback model $\hat{\theta}_F$).

A second parameter λ is also used to control whether word estimates for the feedback model are derived entirely from the set of feedback documents ($\lambda = 0$), the collection language model ($\lambda = 1$), or somewhere in between ($0 < \lambda < 1$). This can be used to promote words which are common in the feedback documents, but not common according to the collection language model.

In these experiments, we use the KL-divergence language model with the absolute discounting method of smoothing with $\Delta = 0.7$. We focus primarily on the two-component mixture model to estimate word probabilities in the feedback model because initial experiments showed this gave higher retrieval performance than the divergence/risk minimisation approach. In all experiments, we stem both query and documents using the Porter stemmer, and remove stopwords using a list of 249 common English words. Images are indexed on *all* caption fields for both the retrieval and feedback indices.

2.3 The Translation Resource: Systran

In these experiments, we translated cross language topics into English using Systran, a popular machine translation (MT) resource which has been used in past CLIR research (see, e.g. [5]). Using Systran as a translation resource, like any form of translation method, can result in erroneous queries because of difficulties encountered during translation including: short queries resulting in little if none syntactic structure to exploit, errors in the original cross language text (e.g. spelling mistakes or incorrect use of diacritics), lack of coverage by the translation lexicon, incorrect translation of phrases, mis-translation of proper names, and incorrect translation of ambiguous words (e.g. selecting the wrong sense of a noun or verb). The effect of translation errors on retrieval performance for ImageCLEF topics is discussed in [7]. For more information on Systran, see e.g. [13].

2.4 Evaluating Relevance Feedback

To evaluate the effects of relevance feedback on cross language image retrieval, we used the following experimental procedure. First, the cross languages topic titles from the ImageCLEF test collection were translated into English using Systran. Using the translated topic titles, we performed an initial retrieval, the baseline, using the set of *strict intersection* relevance judgments that accompany the test collection. To simulate relevance feedback, we extracted relevant documents from the initial ranked lists to create sets of relevance judgments which contain only those relevant found within rank position n. A proportion of these relevant were used to build a feedback model from which additional query terms were selected. A second retrieval was performed with the updated query model to simulate the situation in which a user selects relevant images from the top n.

At the end of the feedback cycle, the effect of relevance feedback is evaluated by comparing it with initial retrieval performance. When no relevant images appear in the top n, Lemur does not generate any query terms and retrieval after feedback results in an empty answer set and no evaluation scores. This affects retrieval after feedback and makes results appear worse than they actually are. Therefore, rather than evaluate an empty answer set the initial retrieval results are used instead. This means results with no relevant in the top n will have no effect on retrieval performance.

Various approaches to evaluating the effects of relevance feedback have been suggested; most derived from the early work on the SMART system (see, e.g. [14]) and include rank freezing, using a residual collection, and using test and control groups. Methods proposed in previous work attempt to eliminate improvements in retrieval due to re-ranking, rather than finding new relevant documents. Measures such as average precision are based on the number of relevant and rank position, but unless documents used for feedback are eliminated from the results a large increase in performance can be observed due to re-ranking. This is particularly noticeable in feedback over several iterations. In these experiments documents from the feedback set are not disregarded after feedback, but this does not affect the validity of our results because we focus on an evaluation measure based on recall at n not rank position where increases in retrieval performance can only come from new relevant documents found in the top n after feedback. We compute an absolute measure of effectiveness before and after feedback and compare the results from this in our evaluation. Although from the user-perspective documents selected for relevance feedback would appear again in the answer set, it is not uncommon to find image retrieval systems which offer this kind of relevance feedback interface (e.g. COMPASS[2] and WebSeek[3]).

2.5 Evaluation Measures

In these experiments, the goal of relevance feedback is to find further relevant documents, particularly when few relevant documents appear in the top n. We use $n = 100$ as a cut-off point for evaluation because from our experiences in building the ImageCLEF test collection, judging the relevance for a large number of images is feasible. The goal, given that users identify relevant images within the top 100, is to automatically find further relevant images (if any more exist) and bring these into the top 100. In this evaluation we use precision at 100 (P_{100}) which measures the proportion of relevant documents found in the top 100, regardless of position. However, this measure is affected by the total number of relevance assessments in the test collection. This does not affect precision at

[2] COMPASS: http://compass.itc.it/

[3] WebSeek: http://www.ctr.Columbia.edu/WebSEEk/

10 or 20, but in the ImageCLEF test collection 90% of topics have fewer than 100 relevant which causes the P_{100} score to be less than 1, even if all relevant images are found in the top 100.

Therefore a normalised precision at 100 measure is used that normalises P_{100} with respect to the number of relevant documents for each query. This measures the proportion of relevant documents retrieved in the top 100 (i.e. a recall measure), rather than the proportion of the top 100 which are relevant. Given a P_{100} score and a set of relevance judgments for a query Φ (the size given by $\mid \Phi \mid$), the normalised precision at 100 score $P_{norm100}$ is given by:

$$P_{norm100} = \frac{P_{100} \times 100}{min(100, \mid \Phi \mid)} \tag{2}$$

The normalised precision score ranges from 0 indicating no relevant in the top 100, to 1 which indicates either all relevant are in the top 100 (if $\mid \Phi \mid \leq 100$) or that all top 100 documents are relevant (if $\mid \Phi \mid > 100$). We define $P_{norm100} = 1$ as a *good* topic (further retrieval unrequired) and $P_{norm100} = 0$ as a *bad* topic (relevance feedback will be unsuccessful unless the user is willing to go beyond the top 100). Summarised across all topics, we use Mean Average Precision (MAP), recall, average P_{100}, and average $P_{norm100}$ scores.

3 Results and Discussion

3.1 Initial Retrieval Performance

Table 1 summarises retrieval performance for initial retrieval without feedback at $n = 100$ and highlights differences between the various measures used in this evaluation. Also shown are the number of *failed* topics, those which either return no images at all, or no relevant in the top 1000. These topics cannot be helped and require the user to reformulate their initial query, or turn to browsing the collection to find relevant images. The number of topics which can be improved are those which are not classed as good or bad.

Table 1. A summary of initial retrieval performance averaged across all topics

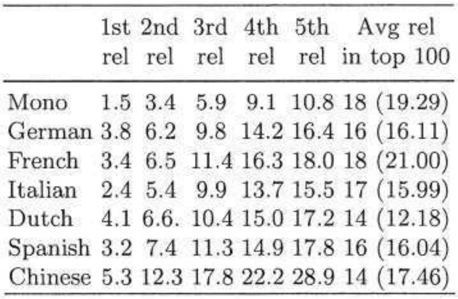
	MAP	%mono MAP	Recall	Avg P_{100}	Avg $P_{norm100}$	%mono $P_{norm100}$	Topics good	Topics bad	Topics to imp.	Topics failed
Mono	0.5514	-	0.8129	0.1800	0.8132	-	22	1	27	1
German	0.4042	73.3%	0.6173	0.1272	0.6450	79.3%	19	9	22	6
French	0.4161	75.5%	0.8489	0.1624	0.6912	85.0%	18	4	28	2
Italian	0.4018	72.9%	0.7898	0.1442	0.6626	81.5%	14	7	29	3
Dutch	0.3806	69.0%	0.7018	0.1134	0.5832	72.0%	15	9	26	4
Spanish	0.3940	71.5%	0.7638	0.1464	0.6504	80.0%	16	4	30	1
Chinese	0.2794	50.7%	0.7255	0.1156	0.5416	67.0%	13	8	29	5

The initial ranking of cross language results is worse than monlingual in the case of all evaluation measures. This is because retrieval is affected by translation errors such as: (1) words other than proper nouns not translated at all (e.g. compound words in German or Dutch), (2) the incorrect translation of words (e.g. in the Italian query "Il monte Ben Nevis", the proper noun "Ben Nevis" is incorrectly translated to give the English query "the mount very Nevis"), (3) the vocabulary of the translation resource not corresponding with the document collection (e.g. the query "road bridge" is translated into "highway bridge" which is not found in the collection), (4) incorrect handling of diacritic characters, and (5) incorrect word sense selection.

According to Table 1, French performs the best out of the cross language results with a MAP score which is 75.5% of monolingual (85% of $P_{norm100}$). We might expect this because Systran has been working longest on the French-English language pair. Other good translations are obtained for German, Italian and Spanish, whereas poorer translation of Chinese and Dutch has clear effects on all evaluation measures. Clearly demonstrated is that each evaluation measure says something different about the results. For example the monolingual recall score indicates that on average 81% of relevant images are returned from all retrieved, average P_{100} that 18% of the top 100 images are relevant, and average $P_{norm100}$ that around 81% of relevant images are retrieved in the top 100.

Also important are the number of good and bad topics. For example in German, although the MAP score is second highest in reality there are 9 topics which do not have any relevant images in the top 100, and 6 of these fail: either they do not return any relevant, or return no results at all (this occurs if German words are not translated and do not match with any index terms). Although topics which do return relevant images have them ranked highly (hence the highest number of good topics after monolingual) the number of bad topics is important because from a user point of view if the system appears to fail for many topics the user will become unsatisfied with their searching regardless how highly images are ranked for other topics.

Table 2. Average rank position (geometric mean) of first 5 relevant images and average number of relevant in top 100 (incl. standard deviation)



	1st rel	2nd rel	3rd rel	4th rel	5th rel	Avg rel in top 100
Mono	1.5	3.4	5.9	9.1	10.8	18 (19.29)
German	3.8	6.2	9.8	14.2	16.4	16 (16.11)
French	3.4	6.5	11.4	16.3	18.0	18 (21.00)
Italian	2.4	5.4	9.9	13.7	15.5	17 (15.99)
Dutch	4.1	6.6.	10.4	15.0	17.2	14 (12.18)
Spanish	3.2	7.4	11.3	14.9	17.8	16 (16.04)
Chinese	5.3	12.3	17.8	22.2	28.9	14 (17.46)

Table 2 shows the average position of the first 5 relevant images for each language and the mean number of relevant in the top 100, ignoring those topics for which $P_{norm100} = 0$. In general relevant images for cross language retrieval appear in lower rank positions than for monolingual and in particular users are going to have to search the furthest in the Chinese and Dutch results. On average, users will find fewer relevant images in cross language retrieval than monolingual, resulting in fewer available images for relevance feedback in these cases. Again, Chinese and Dutch results have least relevant images in the top 100.

3.2 Effect of Parameters on Relevance Feedback

At least four parameters can be varied using `Lemur` for relevance feedback, including: α, λ, the number of relevant documents and the number of selected feedback terms. In this section we show that retrieval performance across language is most affected by α and the number of relevant selected for feedback.

Varying the influence of the collection language model has little effect on retrieval performance using this retrieval system, feedback model and test collection, and these results coincide with the findings of Zhai and Lafferty [12]. Using default settings of 30 terms selected and $\alpha = 0.5$, we find that with 1 feedback document, as λ increases, retrieval performance decreases slightly; with 5 feedback documents the reverse is true, although the variation in retrieval performance is under 2%. In all cases the $P_{norm100}$ score after relevance feedback is always higher than for initial retrieval, and a similar pattern emerges in the cross language results.

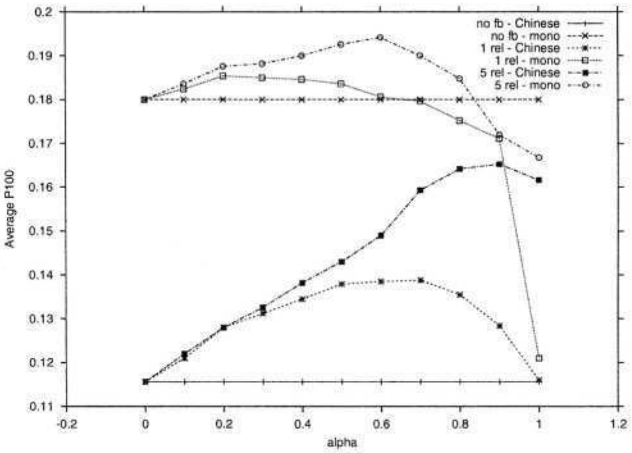
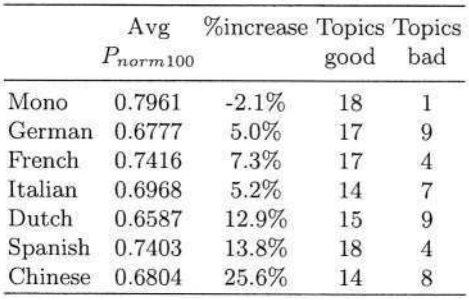


Fig. 2. A plot of average P_{100} for monolingual and Chinese retrieval using 1 or 5 relevant documents for feedback and varying α

Using default settings of $\alpha = \lambda = 0.5$, and selecting between 5 and up to 100 terms for 1 and 5 feedback documents, on average selecting fewer than 30 terms gives lowest performance. With 5 feedback documents retrieval performance reaches its highest with 30 terms; with 1 relevant document a peak is reached with 100 terms (although little variation exists after the selection of 30 terms). The results are similar across language and again, regardless of the number of terms selected, retrieval performance never gets worse.

The parameters which most affect retrieval performance are α and the number of relevant documents used in the feedback model. Fig. 2 shows the average P_{100} score as α is varied for both monolingual and cross language retrieval, and using 1 and 5 relevant documents in the feedback model. Although this coincides with the findings of Zhai and Lafferty, what is interesting are differences across language which result from the quality of the initial query and its influence on term estimates. In the feedback model this means that when the initial query is good (e.g. monolingual) it should have more influence on term weights. In Fig. 2 with 1 feedback document, retrieval performance is highest with $\alpha = 0.2$, whereas with 5 documents, $\alpha = 0.6$ is highest as the feedback model gives better term estimates. Retrieval performance goes below the baseline particularly when $\alpha = 1$ and the initial query is ignored (see Table 3), but the case is very different for other languages, especially Chinese.

Table 3. Results after relevance feedback when ignoring the initial query from the feedback model ($\alpha = 1$) and selecting 30 terms from up to 10 relevant documents



	Avg $P_{norm100}$	%increase	Topics good	Topics bad
Mono	0.7961	-2.1%	18	1
German	0.6777	5.0%	17	9
French	0.7416	7.3%	17	4
Italian	0.6968	5.2%	14	7
Dutch	0.6587	12.9%	15	9
Spanish	0.7403	13.8%	18	4
Chinese	0.6804	25.6%	14	8

In Chinese retrieval, the initial query tends to be worse and therefore larger values of α give higher results than for monolingual. Given 1 feedback document, $\alpha = 0.7$ is highest, whereas for 5 documents $\alpha = 0.9$ is highest. What is most surprising is that even if the initial query is ignored and excluded from the query after feedback ($\alpha = 1$) retrieval performance is still above the baseline, especially when more relevant are used to estimate term probabilities. Upon inspection we find a possible cause for this is that initial monolingual query terms are often generated from the feedback model in the cross language case when the initial query is not effective. In these cases discarding the initial query makes little difference and in fact results in high improvements in $P_{norm100}$ scores over initial retrieval as shown in Table 3. Fig. 3 shows for Chinese retrieval that $\alpha > 0.7$ will give highest results. This trend is similar across language, especially for those languages in which initial retrieval is poor because of translation quality, e.g. Dutch and Italian.

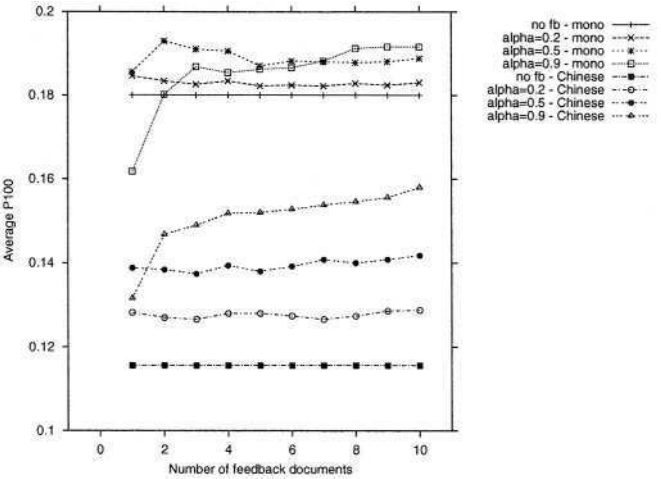


Fig. 3. A plot of average P_{100} for monolingual and Chinese retrieval varying α and using up to 1 to 10 relevant images in the feedback model

Fig. 3 shows the effects of varying the number of feedback documents between 1 and up to 10 (whether the maximum number of relevant are found will be language and topic dependent) at various values of α, with $\lambda = 0.5$ and 30 feedback terms selected for query expansion. As before, the results reflect the effects of α on retrieval performance in both monolingual and cross language for Chinese retrieval. However, in the monolingual case retrieval performance tends to level out and even decreases with $\alpha = 0.5$ because the initial retrieval is actually already high and cannot be improved even with more relevant documents found by the user. For $\alpha = 0.9$, retrieval performance levels out after 8 feedback documents.

In the cross language case, again the results are different. For $\alpha = 0.9$, because emphasis is placed on the relevant documents to estimate the query model, the more images judged relevant by the user, the better. For $\alpha = 0.5$, results are relatively stable and it would seem that the number of feedback documents has little effect on retrieval performance. Generally more feedback documents improve word estimates in the feedback model because terms which better generalise the topic concept are promoted over more specific terms emphasised with fewer relevant documents. For most topics, this tends to be reflected in the query terms where more numbers, dates and proper names are generated. Using more feedback documents also tends to improve the discrimination between which feedback words are related to the topic, and those acting as "noise".

3.3 Relevance Feedback across Language

This section shows the degree of increase one might expect in retrieval performance for each language in the situations where the user selects just 1 relevant image, and up to 10 relevant from the top 100. The results are shown in Tables 4 and 5 where $\alpha = \lambda = 0.5$ and 30 feedback terms are selected from the feedback model. Because we know that results will change depending on α, we also show the maximum increase of $P_{norm100}$ obtained by testing all values of α between 0.1 and 0.9. These settings are, of course, dependent on the ImageCLEF test collection, topics and our retrieval system, but they do provide an idea of the kind of performance increase one could obtain if automatic parameter selection was used.

Table 4. A summary of retrieval performance with relevance feedback from selecting 30 terms from up to 10 relevant in the top 100 ($\alpha = \lambda = 0.5$)

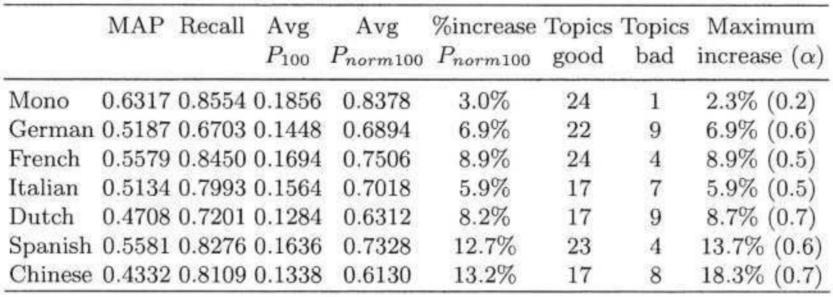
	MAP	Recall	Avg P_{100}	Avg $P_{norm100}$	%increase $P_{norm100}$	Topics good	Topics bad	Maximum increase (α)
Mono	0.6649	0.8356	0.1888	0.8440	3.8%	25	1	4.8% (0.6)
German	0.5831	0.6222	0.1436	0.6848	6.2%	23	9	9.6% (0.8)
French	0.5976	0.8587	0.1816	0.7668	10.9%	24	4	13.5% (0.8)
Italian	0.5765	0.7915	0.1624	0.7096	7.1%	17	7	11.4% (0.8)
Dutch	0.4954	0.6995	0.1304	0.6348	8.9%	17	9	18.1% (0.9)
Spanish	0.5820	0.8172	0.1650	0.7380	13.5%	21	4	19.2% (0.9)
Chinese	0.4806	0.7620	0.1386	0.6170	13.9%	17	8	26.9% (0.9)

The benefits of relevance feedback on retrieval performance are clearly seen in Tables 4 and 5. As expected, when initial retrieval is higher the effects are less dramatic and most benefit is gained for Spanish, Dutch and Chinese retrieval. Relevance feedback, however, does increase both the number of good topics and average $P_{norm100}$ scores in both sets of results. The average increase in performance for $\alpha = \lambda = 0.5$ is similar, but because α is not at its optimum in Table 4, the results are lower than could be obtained if some kind of parameter selection were used. In general more relevant documents are beneficial across all languages (especially non-English), and in the cross language situation, less influence from the initial query in term re-weighting gives better results (i.e. $\alpha \geq 0.8$).

The results in Table 5 show that for most languages retrieval performance can be improved by users selecting just one relevant in the top 100. It would seem that encouraging the user to select either just one or up to 10 relevant in the top 100 (feasible in a CL image retrieval task) would offer substantial benefits for the user and enable more relevant images to be shown to them in the top 100. The results also highlight the differences between the evaluation measures after relevance feedback. For example using Chinese topics the MAP score increases by around 72% but recall only increases by 5%. This indicates the dramatic

MAP increase is caused by re-ranking. The P_{100} increase is around 20% and better indicates the improvement from more relevant being found. However the score is affected by the number of relevance assessments and therefore the most representative measure is $P_{norm100}$ which only increases when more relevant are introduced in the top 100 and indicates that after feedback around 62% of relevant are found in the top 100.

Table 5. A summary of retrieval performance with relevance feedback from selecting 30 terms from 1 relevant in the top 100 ($\alpha = \lambda = 0.5$)



	MAP	Recall	Avg P_{100}	Avg $P_{norm100}$	%increase $P_{norm100}$	Topics good	Topics bad	Maximum increase (α)
Mono	0.6317	0.8554	0.1856	0.8378	3.0%	24	1	2.3% (0.2)
German	0.5187	0.6703	0.1448	0.6894	6.9%	22	9	6.9% (0.6)
French	0.5579	0.8450	0.1694	0.7506	8.9%	24	4	8.9% (0.5)
Italian	0.5134	0.7993	0.1564	0.7018	5.9%	17	7	5.9% (0.5)
Dutch	0.4708	0.7201	0.1284	0.6312	8.2%	17	9	8.7% (0.7)
Spanish	0.5581	0.8276	0.1636	0.7328	12.7%	23	4	13.7% (0.6)
Chinese	0.4332	0.8109	0.1338	0.6130	13.2%	17	8	18.3% (0.7)

A variation in retrieval performance occurs across topics within each language indicating that similar to previous research (e.g. [2]) improvements using relevance feedback are topic dependent. For example topics which have the highest increases in Chinese retrieval include 9, 14 and 15 and tend to be those in which the initial query is poor due to translation errors. For example, topic 15 (Tay bridge rail disaster) is translated from Chinese by Systran as "peaceful bridge railroad disaster" and initially performs badly. After relevance feedback, even with 1 relevant, important discriminating terms such as "Tay" and "rail" are added to the query. Generally the effect of more relevant is to improve retrieval performance (e.g. topic 3) because of the addition of more topic-specific terms, however there are topics when the situation is reversed (e.g. topic 50).

4 Conclusions and Future Work

In this paper, we have explored the use of relevance feedback in cross language image retrieval by simulating user involvement based on relevance assessments from the ImageCLEF test collection. We have shown that relevance feedback can improve retrieval performance across all languages (including monolingual) using the Lemur feedback model and ImageCLEF topics for an ad hoc retrieval task. We have shown in these experiments that on average retrieval performance ($P_{norm100}$) can be improved by as much as 13.9% (for Chinese) and with parameter selection could increase to 26.9%.

As a CL task, we find cross language image retrieval exhibits the following characteristics for relevance feedback: (1) poor initial retrieval due to translation errors, (2) feedback documents will tend to appear in lower rank positions, and (3) fewer relevant images are available for than in the monolingual case. Despite these issues, however, image retrieval is a CL problem in which it is feasible that users are willing and able to browse through many images to find relevant ones even when they are unable to understand the language of the captions because judgments are commonly based on the image itself which is language independent.

Using the two-component mixture feedback model in Lemur, the model is found to behave similar to previous results in the monolingual case, but differently across languages. In cases where translation tends to be worse, e.g. for Chinese, we find that the influence of the initial query should be reduced when calculating term weights in the feedback model. We also find that the more documents used to build the feedback model, the better word estimates are which result in more topics improving in retrieval performance, and reducing those which decrease. The number of feedback terms and the influence of the collection model on term estimates has minimal effects on retrieval for this task.

We have shown how results differ based on different evaluation measures and in particular how MAP can be affected by rank position and appear artificially high after relevance feedback due to effects from re-ranking. We believe that the normalised precision at 100 score offers a useful evaluation measure for this task as it captures the realistic scenario where the user wants to find as many relevant images as possible within the top 100 and rank position is unimportant. The normalised measure is not affected by the number of relevance judgments thereby providing a simple metric that can easily be interpreted.

As an initial investigation the results are encouraging enough to suggest that text-based relevance feedback can play a useful and important role when designing a CL image retrieval system. Given the improvements demonstrated on the ImageCLEF test collection, we would recommend system designers consider how users can be both encouraged to interact with their system and provide feedback. In addition, encouraging users to identify more relevant images seems beneficial across language which could be achieved through improving initial retrieval, or more feedback iterations which also requires a supporting user interface. A final problem is how to deal with failed queries, particularly in a cross language system where queries fail because of translation errors. One solution could be to offer browsing to encourage users to continue pursuing their initial search and find at least 1 relevant which would enable query expansion through relevance feedback to be used.

In future work, we aim to pursue a number of directions including: investigate whether results are improved by using a subset of caption fields, whether removing domain stopwords (i.e. “noise”) can improve results, the effects of users selecting irrelevant documents for feedback (particularly important in image retrieval where relevance assessment is highly subjective), and whether all relevant images used for feedback give similar retrieval performance. This last point addresses topics which are general, e.g. “Postcards of London”, where relevant images are likely to exhibit captions which vary in content (e.g. pictures of famous London sights, locations, people etc.). We would also like to investigate the effects of content-based retrieval methods in relevance feedback as a complement to the text-based approaches and establish whether are findings generalise across different retrieval systems and CL image collections.

Acknowledgements. This work was carried out within the Eurovision project at Sheffield University funded by the EPSRC (Eurovision: GR/R56778/01).

References

1. Efthimiadis, E., Robertson, S.: Feedback and interaction in information retrieval. In Oppenheim, C., ed.: Perspectives in Information Management, London, Butterworths (1989) 257–272
2. Harman, D.: Relevance feedback revisited. In: Proceedings of SIGIR1992. (1992) 1–10
3. Mitra, M., Singhal, A., Buckley, C.: Improving automatic query expansion. In: Proceedings of SIGIR1998. (1998) 206–214
4. Magennis, M., van Rijsbergen, C.J.: The potential and actual efectiveness of interactive query expansion. In: Proceedings of SIGIR1997. (1997) 324–332
5. Flank, S.: Cross-language multimedia information retrieval. In: Proceedings of Applied Natural Language Processing and the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL2000). (2000)
6. Clough, P., Sanderson, M.: The CLEF cross language image retrieval track. In: Submission, to appear. (2003)
7. Clough, P., Sanderson, M.: Assessing translation quality for cross language image retrieval. In: Submission, to appear. (2003)
8. Ballesteros, L., Croft, B.: Resolving ambiguity for cross-language retrieval. In: Proceedings of SIGIR1998. (1998) 64–71
9. McNamee, P., Mayfield, J.: Comparing cross-language query expansion techniques by degrading translation resources. In: Proceedings SIGIR2002. (2002) 159–166
10. Oard, D.: Serving users in many languages. D-Lib magazine (1997)
11. Zhai, C., Lafferty, J.: A study of smoothing methods for langauge models applied to ad hoc information retrieval. In: Proceedings of SIGIR'2001. (2001) 334–342
12. Zhai, C., Lafferty, J.: Model-based feedback in the kl-divergence retrieval model. In: Tenth International Conference on Information and Knowledge Management (CIKM2001). (2001) 403–410
13. Hutchins, W., Somers, H.: An Introduction to machine Translation. Academic Press, London, England (1986)
14. Salton, G.: The SMART Retrieval System - Experiments in Automatic Document Processing. Prentice-Hall Inc., Englewood Cliffs, N.J. (1971)

NN^k Networks for Content-Based Image Retrieval

Daniel Heesch and Stefan Rüger

Department of Computing, Imperial College
180 Queen's Gate, London SW7 2BZ, England
{daniel.heesch,s.rueger}@imperial.ac.uk

Abstract. This paper describes a novel interaction technique to support content-based image search in large image collections. The idea is to represent each image as a vertex in a directed graph. Given a set of image features, an arc is established between two images if there exists at least one combination of features for which one image is retrieved as the nearest neighbour of the other. Each arc is weighted by the proportion of feature combinations for which the nearest neighour relationship holds. By thus integrating the retrieval results over all possible feature combinations, the resulting network helps expose the semantic richness of images and thus provides an elegant solution to the problem of feature weighting in content-based image retrieval. We give details of the method used for network generation and describe the ways a user can interact with the structure. We also provide an analysis of the network's topology and provide quantitative evidence for the usefulness of the technique.

1 Introduction

The problem of retrieving images based not on associated text but on visual similarity to some query image has received considerable attention throughout the last decade. With its origins in computer vision, early approaches to content-based image retrieval (CBIR) tended to allow for little user interaction but it has by now become clear that CBIR faces a unique set of problems which will remain insurmountable unless the user is granted a more active role. The image collections that are the concern of CBIR are typically too heterogenous for object modelling to be a viable approach. Instead, images are represented by a set of low-level features that are a long way off the actual image meanings. In addition to bridging this semantic gap, CBIR faces the additional problem of determining which of the multiple meaning an image admits to is the one the user is interested in. This ultimately translates into the question of which features should be used and how they should be weighted relative to each other. Relevance feedback has long been hailed as the cure to the problem of image polysemy. Although the performance benefits achieved through relevance feedback are appreciable, there remain clear limitations. One of these is the fast convergence of performance during the first few iterations (e.g. [15], [12]), typically halfway before reaching the global optimum. Also, positive feedback, which turns out to be the

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 253–266, 2004.

most efficient feedback method when the collection contains a sufficiently large number of relevant objects, becomes ineffective if the first set of results does not contain *any* relevant items. Not surprisingly, few papers that report performance gains through relevance feedback use collections of sizes much larger than 1000. Possibly as a response to this limitation, research into the role of negative examples has recently intensified (eg [11], [15]). The general conclusion is that negative examples can be important as they allow the user to move through a collection. [15] concludes that negative feedback "offers many more options to move in feature space and find target images. [...] This flexibility to navigate in feature space is perhaps the most important aspect of a content-based image retrieval system."

We would like to take this conclusion further and claim that in the case of large image collections, it becomes absolutely vital to endow a system with the most efficient structures for browsing as well as retrieval. Relevance feedback on negative examples is arguably one possibility but is relatively inefficient if browsing is a main objective. Motivated by these shortcomings of the traditional query-by-example paradigm and of relevance feedback, this paper proposes a novel network structure that is designed to support image retrieval through browsing. The key idea is to attack polysemy by exposing it. Instead of computing at runtime the set of most similar images under a particular feature regime, we seek to determine the set of images that could potentially be retrieved using any combination of features. We essentially determine the union over all feature combinations of the sets of top ranked images. This is done taking each image of the collection in turn as a query. For each image we store the set of images that were retrieved top under some feature regime and the number of times this happened. The latter number provides us with a measure of similarity between two images. Because nearest neighbourhood need not be reciprocated, the similarity measure is asymmetric and the resulting network a directed graph. We refer to the resulting structure as an NN^k network (NN for nearest neighbour and k for the number of different feature types). As it is entirely precomputed, the network allows interaction to take place in real time regardless of the size of the collection. This is in contrast to query-by-example systems, where the time complexity for retrieval is typically linear in the size of the image collection. The storage requirements for the network increase linearly with the number of images. The time complexity of the network generation algorithm is linear in the number of images and at most quadratic in the number of features. In practice, however, the number of features is constant and, as we will show, does not need to be very large to give respectable results.

Using collections of varying size (238, 6129, 32318), we found that the resulting networks have some interesting properties which suggest that the structures constitute 'small-world' networks [21] at the boundary between randomness and high regularity that should make them ideal for organizing and accessing image collections.

The paper is structured as follows: In section 2, we review work that is related to, or has inspired, the technique here introduced. In section 3, we provide details of how the network structure is generated. Section 4 describes the ways a user can interact with the browsing structure. Section 5 presents an analysis of the topological properties of the network and section 6 reports on a quantitative performance evaluation of the network. We conclude the paper in section 7.

2 Related Work

The idea of representing text documents in a nearest neighbour network first surfaced in [7]. The network was, however, strictly conceived as an internal representation of the relationships between documents and terms. The idea was taken up in a seminal paper by Cox ([5] and in greater detail in [6]) in which the nearest neighbour network was identified as an ideal structure for interactive browsing. Cox is concerned with structured databases and envisages one nearest neighbour network for each field of the database with individual records allowing for interconnections between the sets of networks.

Notable attempts to introduce the idea of browsing into CBIR include Campbell's work [3]. His ostensive model retains the basic mode of query based retrieval but in addition allows browsing through a dynamically created local tree structure. The query does not need to be formulated explicitly but emerges through the interaction of the user with the image objects. When an image is clicked upon, the system seeks to determine the optimal feature combination given the current query and the query history, i.e. the sequence of past query images. The results are displayed as nodes adjacent to the query image, which can then be selected as the new query. The emphasis is on allowing the system to adjust to changing information needs as the user crawls through the branching tree.

Jain and Santini's "El niño" system [18] and [17] is an attempt to combine query-based search with browsing. The system displays configurations of images in feature space such that the mutual distances between images as computed under the current feature regime are, to a large extent, preserved. Feedback is given similar as in [11] by manually forming clusters of images that appear similar to the user. This in turn results in an altered configuration with, possibly, new images being displayed.

Network structures that have increasingly been used for information visualization and browsing are Pathfinder networks (PFNETs) [8]. PFNETs are constructed by removing redundant edges from a potentially much more complex network. In [9] PFNETs are used to structure the relationships between terms from document abstracts, between document terms and between entire documents. The user interface supports access to the browsing structure through prominently marked high-connectivity nodes. An application of PFNETs to CBIR is found in [4] where PFNETs are constructed and compared with three different classes of image features (colour, layout and texture) using the similarity between images as the edge weight. According to the authors, the principal strength of the network is its ability to expose flaws in the underlying feature extraction algorithm and the scope for interaction is negligible.

What distinguishes our approach from all previous approaches is the rationale underlying and the method used for network generation, as well as a new notion of similarity between images. In contrast to Cox's networks [5], we determine the nearest neighbour for every combination of features; it is this integration over features that endows the structure with its interesting properties. Also, unlike Pathfinder networks, we do not prune the resulting network but preserve the complete information. This seems justified as we are not concerned with visualizing the entire structure but with facilitating user interaction locally.

3 Network Generation

Given two images X and Y, a set of features, and a vector of feature-specific similarities F, we compute the overall similarity between X and Y as the weighted sum over the feature-specific similarities, i.e.

$$S(X,Y) = \mathbf{w}^{\mathbf{T}}\mathbf{F}$$

with the convexity constraint $|\mathbf{w}|_1 = \sum w_i = 1$ and $w_i \geq 0$. Each of the components F_i represent the similarity between X and Y under one specific feature i which itself can be a complex measure such as shape or colour similarity. According to our construction principle, an image X is connected to an image Y by a directed edge $\overrightarrow{XY}$ if and only if Y is the nearest neighbour of X for at least one combination of features, i.e. if and only if there is at least one instantiation of the weight vector w such that it causes the image Y to have the highest similarity $S(X,Y)$ among all images of the collection (excluding X). Because the overall similarity is a linear sum, small changes in any of the weights will induce correspondingly small changes in the similarity value. Points that are close in weight space should therefore produce a similar ranking and in particular, the same nearest neighbour. We can think of the weight space as being partitioned into a set of regions such that all weights from the same region are associated with the same nearest neighbour. Figure 1 illustrates this idea in the case of a three-dimensional weight space (for details see caption).

The most systematic way of sampling is to impose a grid on the weight space with a fixed number of grid points along each dimension. Using a recursive algorithm with the following recurrence scheme

$$T(1,g) = g$$
$$T(k,g) = \sum_{i=0}^{g} T(k-1, g-i)$$

and setting k and g initially to the number of dimensions and the number of gridpoints along each axis, respectively, we include all permissible gridpoints.

According to this sampling scheme, an image could have more than one thousand nearest neighbours using five features and a grid size of 0.1. In practice, however, the number of distinct neighbours is much smaller and rarely exceeds 50.

The resolution of the grid that is required to capture all nearest neighbours, therefore, is relatively low. Moreover, lacking any additional information, a nearest neighbour that corresponds to a large volume in weight space may reasonably

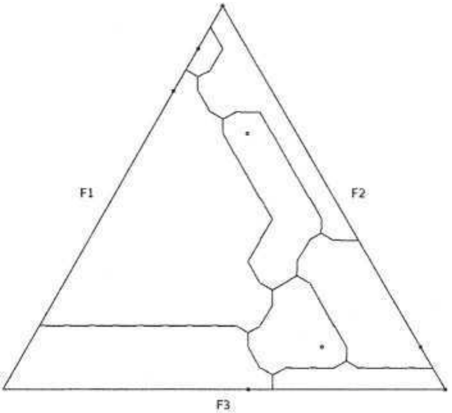


Fig. 1. Simplex showing the partitioning of the weight space into distinct regions for one particular query image. The weights of each of the three features $F1$, $F2$ and $F3$ increase with distance to the corresponding base of the triangle. Each of the bounded regions comprise all those weight sets for which the query has the same nearest neighbour. The points denote the weight combination for each region for which the nearest neighbour had minimum distance to the query.

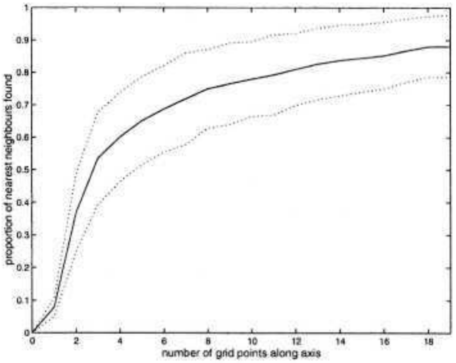


Fig. 2. The proportion of nearest neighbours found for a given grid size averaged over fifty queries (dotted lines: one standard deviation). The exact number of nearest neighbours (100%) for a given query is estimated using 100 gridpoints along each dimension.

be considered more important than one the grid search misses. Figure 2 shows how the number of nearest neighbours rapidly approaches the exact number as the grid size becomes smaller.

It is important to stress, that although, technically, the number of sampled grid points grows exponentially with the dimensionality of the weight space, i.e. the number of features, in practice this number is fixed and limited. Few CBIR applications use more than 10 features. As an illustration, using 7 features and a grid size of 5 per axis, we have a total of 210 grid points to sample.

Using a collection of 32000 images, this can be done in around 50 hours on a standard home computer. With more sophisticated sampling algorithms (such as hierarchical refinement sampling) and parallelization, network construction should be no performance bottleneck even for high-dimensional feature spaces.

For each image we store the set of its nearest neighbours. For each nearest neighbour we also store the proportion of feature combinations in the grid for which that image was ranked top. This number becomes our measure of similarity between two images.

4 Accessing the Network

In order to allow searches without without formulating any query, we provide the user with a representative set of images from the collection by clustering high-connectivity nodes and their neighbours up to a certain depth. Clustering is achieved using the Markov chain clustering (MCL) algorithm [20]. The algorithm reduces the adjacency matrix of the directed graph to a stochastic matrix whose entries can be interpreted as the transition probabilities of moving from one image to another. These probabilities are iteratively updated through an alternating sequence of matrix multiplications and matrix expansions, which have the effect of strengthening frequently used edges. The algorithm has robust convergence properties and allows one to specify the granularity of the clustering. The clustering can be performed offline and may therefore involve the entire image collection. The high sparsity of the adjacency matrix makes the MCL algorithm suitable for even very large networks using sparse matrix techniques.

The interface with the clustering result is shown in Figure 3. We aim to minimize overlap between images while at the same time preserving the cluster structure. The user may select any of the images as a query or as the entry point into the network. Clicking on an image moves it into the center and results in a display of its nearest neighbours. If the size of the set is above a certain threshold the actual number of images displayed is reduced. This threshold T depends on the current size of the window and is updated upon resizing the window. This adjustment is desirable in order to be able to accommodate different screen sizes and makes the system work gracefully with networks that have large variability in the connectivity of its constituent nodes. The criterion for removing images from the set of nearest neighbours is the weight of the arc by which it is connected to the central image (i.e. the area in weight space for which this image is top ranked), only the T images with the highest edge weights are displayed. The neighbours are displayed such that their distances to the central node is a measure of the strength of the connecting edges. The arrangement is found by simulating the evolution of a physical network with elastic springs connecting adjacent nodes.

Through the set of buttons at the bottom of each image, the user can either add images to a query panel (Q) positioned on the left hand side of the display (these images can then be used to start a traditional query-by-example run on the collection), or collect interesting images on a separate panel (A).

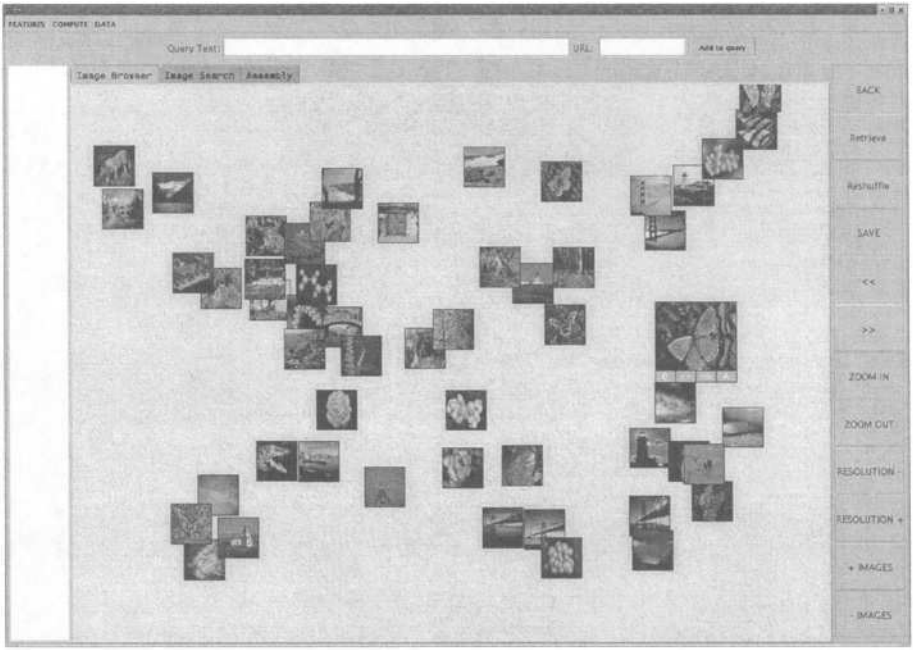


Fig. 3. Initial interface in browsing mode. Displayed are the clusters as determined by the Markov Chain Clustering algorithm. Images become larger when moving the mouse over them.

5 Topological Analysis

5.1 Small-World Properties

An interesting and significant feature of the resulting structure is the presence of so-called small-world properties [21]. Small-world graphs are characterized by two topological properties, both of which are relevant in the context of information retrieval: (i) the clustering coefficient and (ii) the average distance between nodes.

Following Watts and Strogatz [21], one of the basic properties of graphs is the clustering coefficient C. It measures the extent to which a vertex' neighbours are themselves neighbours. More formally, given a graph G without loops and multiple edges and a vertex v, the local clustering coefficient at v is given by the ratio of the number of edges between neighbours of v and the maximum number of such edges (given by $\binom{d_G(v)}{2}$ where $d_G(v)$ is the vertex outdegree of v in G). The clustering coefficient is then obtained by averaging the local clustering coefficient over all vertices. We can think of the clustering coefficient as a measure of the randomness of the graph. It attains a maximum in regular lattice graphs and decreases as we replace edges in the regular graph by randomly positioned edges ([21], [13]). A high clustering coefficient seems, prima facie, to be best suited for the task of information retrieval. However, the

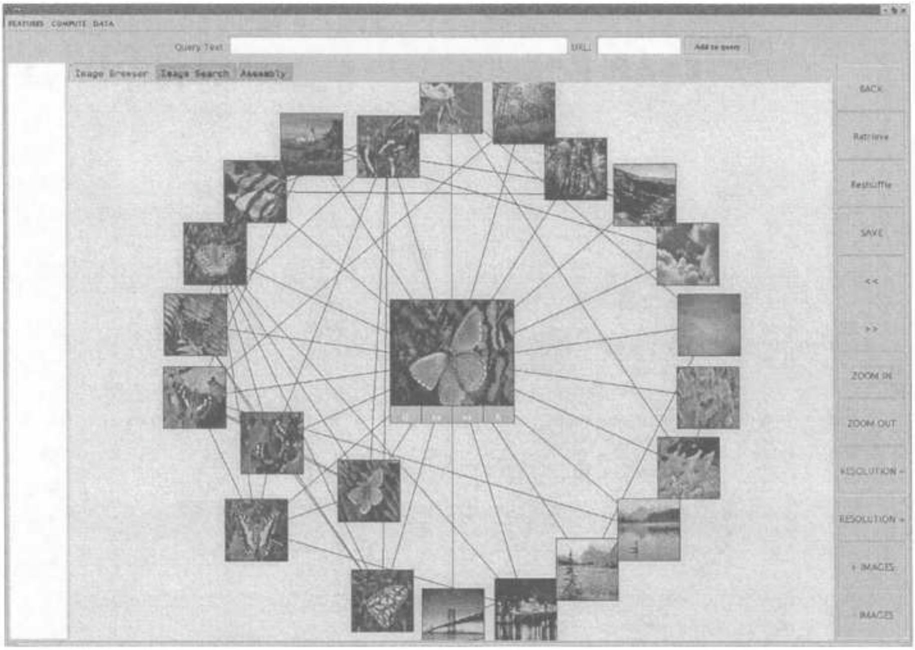


Fig. 4. Local network around the chosen butterfly image depicted in the centre

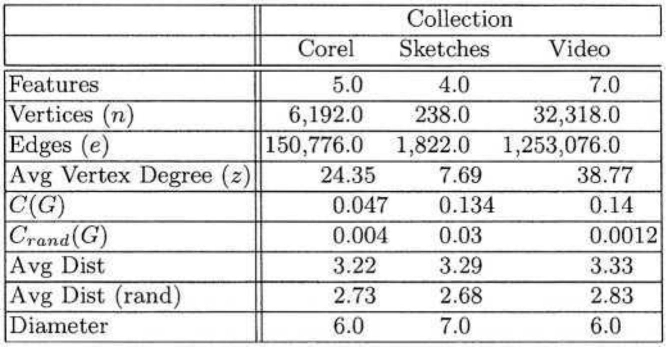
more organized the structure the more difficult it becomes to efficiently move to different areas of the network. Moreover, the simple organization principle that underlies a lattice graph seems inadequate to capture the semantic richness and ambiguity of images. For the purpose of information retrieval, therefore, it appears desirable to have the information organized in structures that are inbetween the two extremes of regularity and randomness.

We have evaluated the clustering coefficients and average distances for three different collections with different feature sets and sizes varying from 238 to 32,318 images (= number of vertices in the network). The clustering coefficient can easily be compared to what would be expected for a random graph. For the classic Erdös-Rényi graph, the expected clustering coefficient is given by z/n where z is the average vertex degree of a graph with n vertices [16]. Likewise, the average distance in a random graph can be approximated by $l = \log(n)/\log(z)$ with n and z as before [2]. For all the three collections examined, the path length is very close to the result of the random graph model while the clustering coefficient exceeds the predicted value by magnitudes, suggesting that the network has indeed a high degree of local structure. The results are summarized in Table 1.

5.2 Degree Distribution

It is of particular interest to see whether the vertex degree sequence is scale-invariant. A large number of distributed systems from social over communication to biological networks display a power-law distribution in their node degree,

Table 1. Analysis of network structure for three different collections. $C(G)$ and $C_{rand}(G)$ denote the clustering coefficients for, respectively, the actual network and a random network with the same number of vertices and edges. The diameter is the largest distance between any two vertices and thus provides an additional measure of the graph's connectivity.



	Collection		
	Corel	Sketches	Video
Features	5.0	4.0	7.0
Vertices (n)	6,192.0	238.0	32,318.0
Edges (e)	150,776.0	1,822.0	1,253,076.0
Avg Vertex Degree (z)	24.35	7.69	38.77
$C(G)$	0.047	0.134	0.14
$C_{rand}(G)$	0.004	0.03	0.0012
Avg Dist	3.22	3.29	3.33
Avg Dist (rand)	2.73	2.68	2.83
Diameter	6.0	7.0	6.0

reflecting the existence of a few nodes with very high degree and many nodes with low degree, a feature which is absent in random graphs. While initial work on scale-free graphs was concerned with investigating their properties and developing generative models, an issue which has only very recently been looked at and which is of relevance to CBIR is the problem of search in these networks when little information is available about the location of the target [1]. Analysis of the degree distributions of the directed graphs constructed thus far suggests that they are, across a broad range of node degrees, scale-free. Figure 5 depicts the frequency distribution of the in-degrees for the network of the video key frame collection (32,318 images). Note that we use the log-scale along the y-axis. If the relationship were of the form $y = e^{-ax+b}$ and thus corresponded to a power-law distribution, the logarithmic plot would reveal this as a straight line $\ln y = -ax + b$. It is typical for such distributions that their boundedness on one or both sides cause the power-law relationship to break down at the boundaries. So in this case, where the number of nodes with exceedingly few neighbours is in fact very small. For a large range of node degrees, however, the relative frequencies seem fairly well described by a power-law distribution.

6 TRECVID 2003 Evaluation

TRECVID (previously the video track of TREC) provides a rare opportunity for research groups in content-based video retrieval to obtain quantitative performance results for realistic search tasks and large image collections. The search task in 2003 involved 24 topics, each exemplified by a set of images and a short text. For each topic, the task was to find the most similar shots and to submit a list with the top ranked 1000 images. Any type of user interaction was allowed after the first retrieval but time for the search was limited to 15 minutes for each topic.

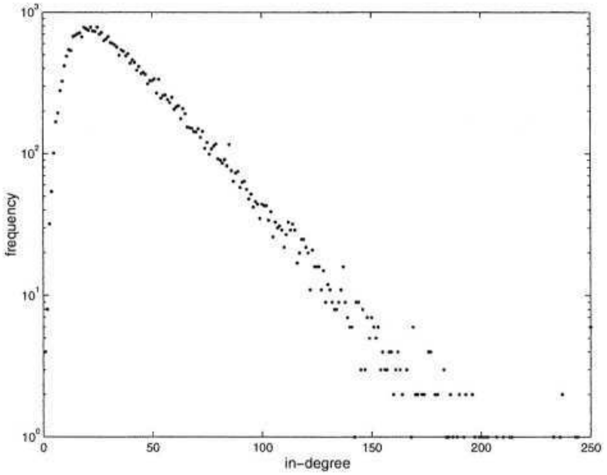


Fig. 5. In-degree distribution for the NN^k network constructed for the video key frame collection

6.1 Features

The NN^k network for the search collection was constructed using seven low-level colour and texture features as well as text from the video transcripts. For the simple texture features, we decided to partition the images into tiles and obtain features from each tile individually with the aim of better capturing local information. The final feature vector for these features consisted of a concatenation of the feature vector of the individual tiles. What follows is a detailed description of each of the features.

HSV Global Colour Histograms: HSV is a cylindrical colour space with H (hue) being the angular, S (saturation) the radial and V (brightness) the height component. This brings about the mathematical disadvantage that hue is discontinuous with respect to RGB coordinates and that hue is singular at the achromatic axis $r = g = b$ or $s = 0$. As a consequence we merge, for each brightness subdivision separately, all pie-shaped 3-d HSV bins which contain or border $s = 0$. The merged cylindrical bins around the achromatic axis describe the grey values which appear in a colour image and take care of the hue singularity at $s = 0$. Saturation is essentially singular at the black point in the HSV model. Hence, a small RGB ball around black should be mapped into the bin corresponding to *hsv* = (0,0,0), to avoid jumps in the saturation from 0 to its maximum of 1 when varying the singular RGB point infinitesimally. There are several possibilities for a natural subdivision of the hue, saturation and brightness axes; they can be subdivided i) linearly, ii) so that the geometric volumes are constant in the cylinder and iii) so that the volumes of the nonlinear transformed RGB colour space are nearly constant. The latter refers to the property that few RGB pixels map onto a small dark V band but many more to a bright V interval of the same size; this is sometimes called the HSV cone in the literature. We use the HSV model with a linear subdivision into 10 hues, 5 saturation values and 5 *V* values yielding a 205-dimensional feature vector. The HSV colour histogram is normalised so that the components add up to 1.

Colour Structure Descriptor: This feature is based on the HMMD (hue, min, max, diff) colour space and is part of the MPEG-7 standard [14]. The HMMD space is derived from the HSV and RGB spaces. The hue component is the same as in the HSV space, and max and min denote the maximum and minimum among the *R, G,* and *B* values, respectively. The diff component is defined as the difference between max and min. Following the MPEG-7 standard, we quantise the HMMD non-uniformly into 184 bins with the three dimensions being hue, sum and diff (sum being defined as *(max + min)* /2) and use a global histogram.

In order to capture local image structure, we slide a 8 × 8 structuring window over the image. Each of the 184 bins of the colour structure histogram contains the number of window positions for which there is at least one pixel falling into the corresponding HMMD bin. This descriptor is capable of discriminating between images that have the same global colour distribution but different local colour structures. Although the number of samples in the 8 × 8 structuring window is kept constant (64), the spatial extent of the window differs depending on the size of the image. Thus, for larger images appropriate sub-sampling is employed to keep the total number of samples per image roughly constant. The 184 bin values are normalised by dividing by the number of locations of the structuring window; each of the bin values falls thus in the range [0,1], but the sum of the bin values can take any value up to 64 (see [14] for details).

Thumbnail feature: This feature is obtained by scaling down the original image to 44 × 27 pixels and then recording the gray value of each of the pixels leaving us with a feature vector of size 1,188. It is suited to identify near-identical copies of images, eg, key frames of repeated shots such as adverts.

Convolution filters: For this feature we use Tieu and Viola's method [19], which relies on a large number of highly selective features. The feature generation process is based on a set of 25 primitive filters, which are applied to the gray level image to generate 25 different feature maps. Each of these feature maps is rectified and downsampled and subsequently fed to each of the 25 filters again to give 625 feature maps. The process is repeated a third time before each feature map is summed to give 15,625 feature values. The idea behind the three stage process is that each level 'discovers' arrangements of features in the previous level and ultimately leads to a set of very selective features, each of which takes high values only for a small fraction of the image collection. The feature generation process is computationally quite costly, but only needs to be done once.

Variance Feature: The variance feature is a 20 bin histogram of gray value standard deviations within a a sliding window of size 5 × 5 determined for each window position. The histogram is computed for each of 9 non-overlapping image tiles and the bin frequencies concatenated to give a feature vector of size 180.

Uniformity Feature: Uniformity is another statistical texture feature defined as

$$U := \sum_{z=0}^{L-1} p^2(z)$$

where $L = 100$ is the number of gray levels and $p(z)$ the frequency of pixels of gray level z. For each of 8×8 image tiles, we obtain one uniformity value resulting in a feature vector of size 64.

Bag of words: Using the textual annotation obtained from the video transcripts provided, we compute a bag-of-words feature consisting for each image of the set of accompanying stemmed words (Porter's algorithm) and their weights. These weights are determined using the standard tf-idf formula and normalised so that they sum to one. As this is a sparse vector of considerable size (the number of different words) we store this feature in the form of (weight, word-id) pairs, sorted by word-id.

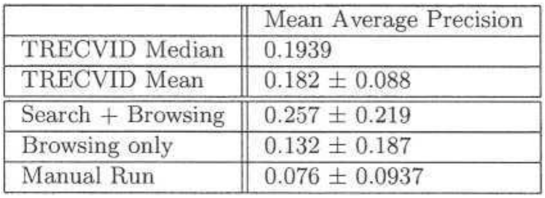
6.2 Distances and Normalisation

In order to compare two images in the collection we use distances of their corresponding features. For these we use the L_1-norm throughout (and the L_1 norm raised to the power of 3 for the bag-of-stemmed-words). Some of the distances already exhibit a natural normalisation, for example when the underlying features are normalised (eg the HSV colour histograms), others do not (eg the colour structure descriptor). As the distances for different features are to be combined, we normalise the distances empirically for each feature, such that their median comes to lie around one.

6.3 Results

Four interactive runs were carried out, in one of which the user was only allowed to find relevant shots by browsing through the NN^k network. For this run text and images were only used to inform about the task. Although our best interactive runs were those that employ a mixture of search, relevance feedback and browsing, the performance (as measured in mean average precision over all 24 topics) of the browsing-only run was considerably better than that of a manual run in which images were retrieved with a fixed set of feature weights and no subsequent user interaction. Performance also proved superior to more than 25% of all the 36 interactive runs submitted by the participating groups, all of which used some form of automatic search-by-exammple. Considering the number of

Table 2. Performance of the browse-only run compared to our interactive search run with browsing and our best manual run with no user interaction and the mean and median of the 36 interactive runs from all groups.



	Mean Average Precision
TRECVID Median	0.1939
TRECVID Mean	0.182 ± 0.088
Search + Browsing	0.257 ± 0.219
Browsing only	0.132 ± 0.187
Manual Run	0.076 ± 0.0937

features and the size of the collection, these results are quite respectable and demonstrate that browsing in general and the proposed structure in particular have a potential for CBIR that should not be left unexploited. A summary of the results is given in Table 2 and more details can be found in [10].

7 Conclusions

The strengths of the proposed structure are twofold: (i) it provides a means to expose the semantic richness of images and thus helps to alleviate the problem of image polysemy which has been for many years a central research concern in CBIR. Instead of displaying all objects that are similar under only one, possibly suboptimal, feature regime, the user is given a choice between a diverse set of images, each of which is highly similar under *some* interpretation, (ii) the structure is precomputed and thus circumvents the often inacceptable search times encountered in traditional content-based retrieval systems. Interaction is in real time, regardless of the collection size.

The NN^k technique presented here is of wider applicability. Its usefulness naturally extends to any multimedia objects for which we can define a similarity metric and a multidimenional feature space, such as text documents or pieces of music. It is, however, in the area of image retrieval that it should find its most profitable application as relevance can be assessed quickly and objects can be displayed at a relatively small scale without impeding object understanding. Although the principal motivation behind the NN^k network is to mitigate the problems associated with category search in large collections, the topology should make it an ideal structure for undirected browsing also.

Acknowledgements. This work was partially supported by the EPSRC, UK.

References

1. L A Adamic, R M Lukose, A R Puniyani, and B A Huberman. Search in power-law networks. *Physical Review E,* 64, 2001.
2. B Bollobás. *Random Graphs.* Springer, New York, 1985.

3. I Campbell. *The ostensive model of developing information-needs.* PhD thesis, University of Glasgow, 2000.
4. C Chen, G Gagaudakis, and P Rosin. Similarity-based image browsing. In *Proceedings of the 16th IFIP World Computer Congress. International Conference on Intelligent Information Processing,* 2000.
5. K Cox. Information retrieval by browsing. In *Proceedings of The 5th International Conference on New Information Technology, Hongkong,* 1992.
6. K Cox. *Searching through browsing.* PhD thesis, University of Canberra, 1995.
7. B Croft and T J Parenty. Comparison of a network structure and a database system used for document retrieval. *Information Systems,* 10:377–390, 1985.
8. D W Dearholt and R W Schvaneveldt. *Properties of Pathfinder networks, In R W Schvaneveldt (Ed.), Pathfinder associative networks: Studies in knowledge organization.* Norwood, NJ: Ablex, 1990.
9. R H Fowler, B Wilson, and W A L Fowler. Information navigator: An information system using associative networks for display and retrieval. *Department of Computer Science, Technical Report NAG9-551, 92-1,* 1992.
10. D Heesch, M Pickering, A Yavlinsky, and S Rüger. Video retrieval within a browsing framework using keyframe. In *Proceedings of TRECVID 2003, NIST (Gaithersburg, MD, Nov 2003),* 2004.
11. D C Heesch and S Rüger. Performance boosting with three mouse clicks — relevance feedback for CBIR. In *Proceedings of the European Conference on IR Research 2003.* LNCS, Springer, 2003.
12. D C Heesch, A Yavlinsky, and S Rüger. Performance comparison between different similarity models for CBIR with relevance feedback. In *Proceedings of the International Conference on video and image retrieval (CIVR 2003), Urbana-Champaign, Illinois.* LNCS, Springer, 2003.
13. J M Kleinberg. Navigation in a small world. *Nature,* page 845, 2000.
14. B S Manjunath and J-S Ohm. Color and texture descriptors. *IEEE Transactions on circuits and systems for video technology,* 11:703–715, 2001.
15. H Müller, W Müller, D M Squire, M.S Marchand-Maillet, and T Pun. Strategies for positive and negative relevance feedback in image retrieval. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR 2000), IEEE, Barcelona, Spain,* 2000.
16. M E J Newman. *Random graphs as models of networks, In S Bornholdt and H G Schuster (Ed.), Handbook of graphs and networks - from the genome to the internet.* Wiley-VCH, 2003.
17. S Santini, A Gupta, and R Jain. Emergent semantics through interaction in image databases. *IEEE transactions on knowledge and data engineering,* 13(3):337–351, 2001.
18. Simone Santini and Ramesh Jain. Integrated browsing and querying for image databases. *IEEE MultiMedia,* 7(3):26–39, 2000.
19. Kinh Tieu and Paul Viola. Boosting image retrieval. In *5th International Conference on Spoken Language Processing,* December 2000.
20. S van Dongen. A cluster algorithm for graphs. *Technical report, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam,* 2000.
21. D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature,* 393:440–442, 1998.

Integrating Perceptual Signal Features within a Multi-facetted Conceptual Model for Automatic Image Retrieval

Mohammed Belkhatir, Philippe Mulhem, and Yves Chiaramella

Laboratoire CLIPS-IMAG,
Université Joseph Fourier, Grenoble, France
{belkhatm,mulhem,chiara}@imag.fr

Abstract. The majority of the content-based image retrieval (CBIR) systems are restricted to the representation of signal aspects, e.g. color, texture... without explicitly considering the semantic content of images. According to these approaches a sun, for example, is represented by an orange or yellow circle, but not by the term "sun". The signal-oriented solutions are fully automatic, and thus easily usable on substantial amounts of data, but they do not fill the existing gap between the extracted low-level features and semantic descriptions. This obviously penalizes qualitative and quantitative performances in terms of recall and precision, and therefore users' satisfaction. Another class of methods, which were tested within the framework of the Fermi-GC project, consisted in modeling the content of images following a sharp process of human-assisted indexing. This approach, based on an elaborate model of representation (the conceptual graph formalism) provides satisfactory results during the retrieval phase but is not easily usable on large collections of images because of the necessary human intervention required for indexing. The contribution of this paper is twofold: in order to achieve more efficiency as far as user interaction is concerned, we propose to highlight a bond between these two classes of image retrieval systems and integrate signal and semantic features within a unified conceptual framework. Then, as opposed to state-of-the-art relevance feedback systems dealing with this integration, we propose a representation formalism supporting this integration which allows us to specify a rich query language combining both semantic and signal characterizations. We will validate our approach through quantitative (recall-precision curves) evaluations.

1 Introduction

From a user's standpoint, the democratization of digital image technology has led to the need to specify new image retrieval frameworks combining expressivity, performance and computational efficiency.

The first CBIR systems (signal-based) [11,16,18,19] propose a set of still images indexing methods based on low-level features such as colors, textures... The general approach consists in computing structures representing the image distribution such as color histograms, texture features and using this data to partition the image; thus

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 267–282, 2004.

reducing the search space during the image retrieval operation. These methods are based on the computation of discriminating features rejecting images which do not correspond to the query image and hold the advantage of being fully automatic, thus are able to quickly process queries. However, aspects related to human perception are not taken into account. Indeed, an image cannot be sufficiently described by its moments or color histograms. The problem arising from invariants or discriminating features lies on the loss of semantic information conveyed by the image. These tools are used for restricting the search space during the retrieval operation but cannot however give a sound and complete interpretation of the content. For example, can we accept that our system considers red apples or Ferraris as being the same entities simply because they present similar color histograms? Definitely not, as shown in [9], taking into account aspects related to the image content is of prime importance for efficient photograph retrieval.

In order to overcome this weakness and allow the representation of the semantic richness of an image, semantic-based models such as Vimsys [5] and EMIR^2 [12,17] rely on the specification of a set of logical representations, which are multilevel abstractions of the physical image. The originality of these models is achieved through integration of heterogeneous representations within a unique structure, collecting a maximum of information related to the image content. However these systems present many disadvantages. First, they are not fully automatic and require the user intervention during indexing, which constitutes a major drawback when dealing with reasonable corpus of images as this process is time-consuming and leads to heterogeneous and subjective interpretations of the image semantic content. Moreover, these models do not incorporate a framework for signal characterization, e.g. a user is not able to query these systems for “red roses”. Therefore, these solutions do not provide a satisfying solution to bridge the gap between semantics and low-level features.

State-of-the-art systems which attempt to deal with the signal/semantics integration [6,10,21] are based on the association of a query by example framework with textual annotation. These systems mostly rely on user feedback as they do not provide a formalism supporting the specification of a full textual querying framework combining semantics and signal descriptions and therefore exhibit poor performance in relating low-level features to high-level semantic concepts. Prototypes such as ImageRover [6] or iFind [10] present loosely-coupled solutions relying on textual annotations for characterizing semantics and a relevance feedback scheme that operates on low-level features. These approaches have two major drawbacks: first, they fail to exhibit a single framework unifying low-level and semantics, which penalizes the performances of the system in terms of retrieval effectiveness and quality. Then, as far as the querying process is concerned, the user is to query both textually in order to express high-level concepts and through several and time-consuming relevance feedback loops to complement her/his initial query. This solution for integrating semantics and low-level features, relying on a cumbersome querying process does not enforce facilitated and efficient user interaction. For instance, queries involving textually both semantics and signal features such as “Retrieve images with a purple flower” or “Retrieve images with a vegetation which is mostly green” cannot be processed. We propose a unified framework coupling semantics and signal features for automatic image retrieval that enforces expressivity, performance and computational efficiency. As opposed to state-of-the-art frameworks offering a loosely-coupled solution with a textual framework for keyword-based querying integrated in a relevance feedback framework operating on low-level features, user interaction is optimized through the specification of a unified textual querying framework that allows to query over both signal and semantics.

In the remainder of this paper, we will first present the general organization of our model and the representation formalism allowing the integration of semantics and signal features within an expressive and multi-facetted conceptual framework. We will deal in sections 3 and 4 with the descriptions of both the semantic and the signal facets, dealing thoroughly with conceptual index structures. Section 5 will specify the querying framework. We finally present the validation experiments conducted on a test collection of 2500 personal photographs.

2 The Proposed Method: Signal/Semantic Integration within an Expressive Conceptual Framework

In state-of-the-art CBIR systems, images cannot be easily or efficiently retrieved due to the lack of a comprehensive image model that captures the structured abstractions, the signal information conveyed and the semantic richness of images. To remedy such shortcomings, visual semantics and signal features are integrated within an image model which consists of a physical image level and a conceptual level. The latter is itself a multi-facetted framework supported by an expressive knowledge representation formalism: conceptual graphs.

2.1 An Image Model Integrating Signal and Semantic Features

The first layer of the image model (fig.1) is the physical image level representing an image as a matrix of pixels.

The second layer is the conceptual level and is itself a tri-facetted structure:

– The first facet called *objectfacet* describes an image as a set of **image objects,** abstract structures representing visual entities within an image. Their specification is an attempt to operate image indexing and retrieval operations beyond simple low-level processes [18] or region-based techniques [2] since image objects convey the visual semantics and the signal information at the conceptual level. Formally, this facet is described by the set $\mathbf{I}_{IO}$ of image object identifiers.

– The second facet called ***visual semantic facet*** describes the image semantic content and is based on labeling image objects with a semantic concept. In the example image of fig. 1, the first image object (Io1) is tagged by the semantic concept *Hut.* Its formal description will be dealt with in section 3.

– The third facet called ***signalfacet*** describes the image signal content in terms of symbolic perceptual features and consists in characterizing image objects with signal concepts. In the example image of fig. 1, the second image object (Io2) is associated with symbolic colors *Cyan* and *White.* The signal facet will be described in detail and formalized in section 4.

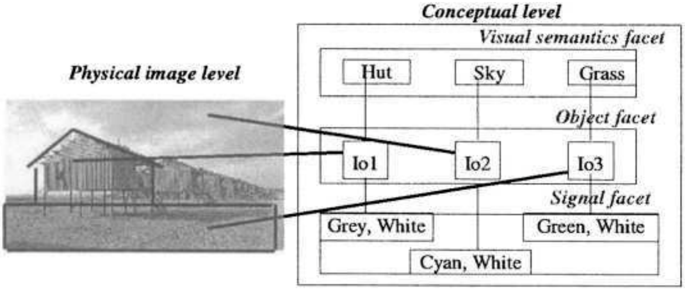


Fig. 1. Image model

2.2 Representation Formalism

In order to instantiate this model within a framework for image retrieval, we need a representation formalism capable to represent image objects as well as the visual semantics and signal information they convey. Moreover, this representation formalism should make it easy to visualize the information related to an image. It should therefore combine expressivity and a user-friendly representation. As a matter of fact, a graph-based representation and particularly conceptual graphs (CGs) [22] are an efficient solution to describe an image and characterize its components. The asset of this knowledge representation formalism is its flexible adaptation to the symbolic approach of image retrieval [12,18,19]. It allows indeed to represent components of our CBIR architecture and to develop an expressive and efficient framework as far as indexing and querying operations.

Formally, a conceptual graph is a finite, bipartite, connex and oriented graph. It features two types of nodes: the first one graphically represented by a rectangle (fig. 2) is tagged by a concept however the second represented by a circle is tagged by a conceptual relation. The graph of fig. 2 represents a man eating in a restaurant. Concepts and relations are identified by their type, itself corresponding to a semantic class. Concept and conceptual relation types are organized within a lattice structure partially ordered by '≤' which expresses the relation 'is a specialization of'. For example, Person ≤ Man denotes that the type *Man* is a specialization of the type *Person,* and will therefore appear in the offspring of the latter within the lattice organizing these concept types. Within the scope of the model, conceptual graphs are used to represent the image content at the conceptual level. Each image (respectively user query) is represented by a conceptual graph called document index graph (respectively query graph) and evaluation of similarity between an image and a query is achieved through a correspondence function: the conceptual graph projection operator.

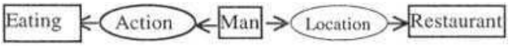


Fig. 2. An example of conceptual graph

3 A Descriptive Model for Semantics: The Visual Semantics Facet

3.1 Conceptual Structures for the Visual Semantics Facet

Semantic concept types are learned and then automatically extracted given a visual thesaurus. The construction of a visual thesaurus is strongly constrained by the application domain, indeed dealing with corpus of medical images would entail the elaboration of a visual thesaurus that would be different from a thesaurus considering computer-generated images. In this paper, our experiments presented in section 6 are based on a collection of personal photographs. Let us detail the elaboration of our visual thesaurus.

Several experimental studies presented in [14] have led to the specification of twenty categories or picture scenes describing the image content at the global level. Web-based image search engines (google, altavista) are queried by textual keywords corresponding to these picture scenes and 100 images are gathered for each query. These images are used to establish a list of concept types characterizing objects that can be encountered in these scenes. This process highlights seven major semantic concept types: people, ground, sky, water, foliage, mountain/rocks and building. We then use WordNet to produce a list of hyponyms linked with these concept types and discard terms which are not relevant as far as indexing and retrieving images from personal corpus are concerned. Therefore, we obtain a set of concept types which are specializations of the seven major semantic concept types. We repeat the process of finding hyponyms for all the specialized concept types. The last step consists in organizing all these concept types within a multi-layered lattice ordered by a specific/generic partial order. In fig. 3, the second layer of the lattice consists of concepts types which are specifications of the major semantic concept types, e.g.*face* and *crowd* are specifications of *people.* The third layer is the basic layer and presents the most specific concept types, e.g. *man, woman, child* are specifications of *individual.*

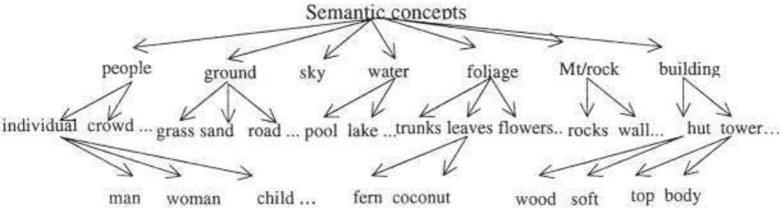


Fig. 3. Lattice of semantic concept types

A feed-forward neural network is used to learn these semantic concept types from a set of training and testing images. Low-level features [8] are computed for each training object and organized within a vector used as the input of the neural network. The learning being completed, an image is then processed by the network and the recognition results are aggregated to highlight image objects. An image object is thus characterized by a vector of semantic concept types, each one being linked to a value of recognition certainty. For instance, in the image of fig. 1, the second image object labeled as Io2 is characterized by a vector which has a significant value for the semantic concept type *hut* and small values related to other semantic concepts. At the

CG representation level, the semantic concept type with the highest recognition certainty is kept. As a matter of fact, Io2 will be represented by the semantic concept type *hut.* We will now specify the model organizing the visual semantics facet and deal with its representation in terms of CGs.

3.2 Model of the Visual Semantics Facet

The model of visual semantics facet gathers semantic concept types and their lattice induced by a partial order relation: Msy = (SC, sct)

- SC is the set of visual semantics concept types.
- sct: $I_{IO} \rightarrow SC$ associates to each image object its semantic concept type.

Image objects are represented by *Io* concepts and the set SC is represented by a lattice of semantic concept types partially ordered by the relation $\leq_{vs}$. An instance of the visual semantics facet is represented by a set of CGs, each one containing an *Io* type linked through the conceptual relation *sct* to a semantic concept type. The basic graph controlling the generation of all visual semantic facet graphs is: [Io]→(sct)→[SC]. For instance, the following graphs are the representation of the visual semantics facet in fig. 1 and can be translated as: the first image object (Io1) is associated with the semantic concept type *hut,* the second image object (Io2) with the semantic concept type *sky* and the third image object (Io3) with the semantic concept type *grass.*

Io1
sct
Hut
Io2
sct
Sky
Io3
sct
Grass

The integration of signal information within the conceptual level is crucial since it enriches the indexing framework and expands the query language with the possibility to query over both semantics and visual information. After presenting our formalism, we will now focus on the signal facet and deal with theoretical implications of integrating signal features within our multi-facetted conceptual model. This integration is not straightforward as we need to characterize low-level signal features at the conceptual level, and therefore specify a rich framework for conceptual signal indexing and querying. We first propose conceptual structures for the signal facet and then thoroughly specify the conceptual model for the signal facet and its representation in terms of CGs.

4 The Signal Facet: From Low-Level Signal Data to Symbolic Characterization

Our architecture and its supported operational model make it possible for a user to combine low-level features with visual semantics in a fully textual conceptual framework for querying. However, querying textually on low-level features requires specifying a correspondence process between color names and color stimuli.

Our symbolic representation of color information is guided by the research carried out in color naming and categorization [1] stressing a step of correspondence between color names and their stimuli. We will consider the existence of a formal system S_{nc} of color categorization and naming [7] which specifies a set of color categories Cat with a cardinal C_{cat}. These color categories are the C_i where variable i belongs to [1, C_{cat}]. Each image object is then indexed by two types of conceptual structures featuring its color distribution: boolean and quantified signal concepts.

When providing examples of the specified conceptual structures, we will consider that the color naming and categorization system highlights four color categories: cyan(c), green(gn), grey(g) and white(w) (Cat={cyan, green, grey, white}).

4.1 Index Structures

Boolean signal index concept types *(BSICs),* gathered within the set B_{SI} are supported by a vector structure v_B with a number of elements equal to the number C_{cat} of color categories highlighted by the naming and categorization system. Values $v_B[i]$, $i \in [1, C_{cat}]$ are booleans stressing that the color category C_i is present in non-zero proportion within the considered image object. The semantics conveyed by BSICs is the 'And' semantics. As a matter of fact, these concept types feature the signal distribution of image objects by a conjunction of color categories. For instance, the first image object (Io1) corresponding to the semantic concept type *hut* in fig.1 is characterized by the BSIC <c:0,gn:0,g:1,w:1>, which is translated by Io1 having a signal distribution including grey **and** white.

We wish to extend the definition of BSICs and quantify by a variable known as **color category value** the integer percentage of pixels corresponding to a color category. The color category value corresponds to the standardization of the pixel percentages of each color category. Quantified signal index concept types *(QSICs),* gathered within the set Q_{SI} are supported by a vector structure v_Q with a number of elements equal to the number C_{cat} of color categories highlighted by the naming and categorization system. Values $v_Q[i]$, $i \in [1, C_{cat}]$ are the color category values. These concept types feature the signal distribution of image objects by a conjunction of color categories and their associated color category values. Let us note that the sum of category values is always 100, the color distribution being fully distributed between all color categories. The second image object (Io2) corresponding to the semantic concept type *sky* in fig.1 is characterized by the QSIC <c:59,gn:0,g:0,w:41>, which is translated by Io2 having a signal distribution including 59% of cyan **and** 41% of white.

4.2 Index Structures

As far as querying is concerned, our conceptual architecture is powerful enough to handle an expressive and computationally efficient language consisting of boolean and quantified queries:

– A user shall be able to associate visual semantics with a boolean conjunction of color categories through an *And* query, such as Q1: "Find images with a grey and white hut", a boolean disjunction of color categories through an *Or* query, such as Q2: "Find images with either cyan or grey sky" and a negation of color categories through a *No* query, such as Q3: "Find images with non-white flowers".

– As far as quantified queries, *At Most* queries (such as Q4: "Find images with a cloudy sky (at most 25% of cyan)") and *At Least* queries (such as Q5: "Find images with lake water (at least 25% of grey)") associate visual semantics with a set of color categories and a percentage of pixels belonging to each one of these categories. We specify also literally quantified queries *(Mostly, Few)* which can prove easier to handle by an average user, less interested in precision-oriented querying.

In the following sections we will present the conceptual structures supporting the previously defined query types.

4.2.1 Boolean signal query concept types. There are three categories of concepts types supporting boolean querying: *And* signal concept types *(ASCs),* gathered within the set B_{And} represent the color distribution of an image object by a conjunction of color categories; *Or* signal concept types *(OSCs),* gathered within the set B_{Or}, by a disjunction of color categories and *No* signal concept types *(NSCs),* gathered within the set B_{No}, by a negation of color categories. These concepts are respectively supported by vector structures v_{And}, v_{Or} and v_{No} with a number of elements equal to the number C_{cat} of color categories. Values $v_{And}[i]$, $v_{Or}[i]$ and $v_{No}[i]$, $i \in [1,C_{cat}]$ are non-null if the color category C_i is mentioned respectively in the conjunction, disjunction or negation of color categories within the query. The ASC corresponding to the color distribution expressed in query Q1 is $<c:0, gn:0, g:1, w:1>_{And}$. The color distribution expressed in query Q2 is translated in the OSC $<c:1, gn:0, g:1, w:0>_{Or}$. Finally, the color distribution expressed in query Q3 is translated in the NSC $<c:0, gn:0, g:0, w:1>_{No}$.

4.2.2 Signal quantified query concept types. There are two types of numerically quantified signal concept types : *At Most* signal concept types *(AMSCs)* gathered within the set Q_{AM} and *At Least* signal concept types *(ALSCs)* gathered within the set Q_{AL} that are respectively supported by vector structures v_{AM} and v_{AL} with a number of elements equal to C_{cat}.

If the color category C_i is specified in a query, values $v_{AM}[i]$ and $v_{AL}[i]$ ($i \in [1,C_{cat}]$) are non-null and correspond respectively to the maximum pixel percentage associated with C_i (translating the keyword 'At Most') and the minimum pixel percentage associated with C_i (translating the keyword 'At Least'). For instance, the color distribution expressed in query Q4 is translated in the AMSC $<c:25, gn:0, g:0, w:0>_{AMSC}$ whereas the color distribution expressed in query Q5 is translated in the ALSC $<c:0, gn:0, g:25, w:0>_{ALSC.}$

Expressing a query with numerical quantification is precision-oriented and an average user might find it cumbersome. Therefore we introduce literally quantified queries such as *Mostly* queries (e.g. "Find images with a bright sky (*Mostly* cyan)") and *Few* queries (e.g. "Find images with a cloudy sky (*Few* cyan)"). These queries are supported by sets Q_{Mostly} and Q_{Few} of *Mostly* and *Few* signal concept types. We set up a correspondence between the quantifier *Mostly* and the numeral quantification *At Least 50%.* Also, the quantifier *Few* will correspond to the numeral quantification *At Most 10%.*

After introducing structures for conceptual signal characterization, we propose a formal model organizing the signal facet. This model is then instantiated in the CG representation formalism within our image retrieval framework.

4.3 A Conceptual Model for the Signal Facet

The model of the signal facet $\mathbf{M_{SI}}$ is given by the model $\mathbf{MI_{SI}}$ of the signal index facet and the model $\mathbf{MQ_{SI}}$ of the signal query facet where $\mathbf{MI_{SI} = (I_{SI}, RI_{SI})}$ and $\mathbf{MQ_{SI} = (Qu_{SI}, RQ_{SI})}$:

- $\mathbf{I_{SI}}$ is the set of signal index structures: $\mathbf{I_{SI} = \{B_{SI}, Q_{SI}\}}$
- $\mathbf{RI_{SI}}$ is the set of signal index conceptual relations: $\mathbf{RI_{SI}}$ = **{b_si, q_si}**

  b_si: $I_{IO} \rightarrow B_{SI}$ and q_si : $I_{IO} \rightarrow Q_{SI}$ associate image object identifiers with boolean and quantified signal index concept types.
- $\mathbf{Qu_{SI}}$ is the set of signal query structures: $\mathbf{Qu_{SI} = \{B_{And}, B_{Or}, B_{No}, Q_{AM}, Q_{AL}, Q_{Mostly}, Q_{Few}\}}$
- $\mathbf{RQ_{SI}}$ is the set of signal query conceptual relations: $\mathbf{RQ_{SI}}$ = **{and_si , or_si, no_si, am_si, al_si, mostly_si, few_si}**

  and_si : $I_{IO} \rightarrow B_{And}$; or_si : $I_{IO} \rightarrow B_{Or}$ and no_si : $I_{IO} \rightarrow B_{No}$ associate image object identifiers with ASCs, OSCs and NSCs.

  am_si : $I_{Si} \rightarrow Q_{AM}$ and al_si : $I_{IO} \rightarrow Q_{AL}$ associate image object identifiers with AMSCs and ALSCs.

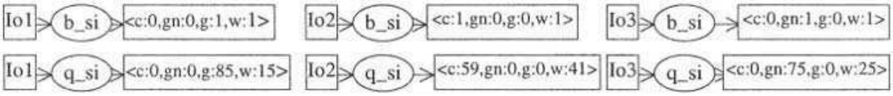
  mostly_si : $I_{IO} \rightarrow Q_{Mostly}$ and few_si : $I_{IO} \rightarrow Q_{Few}$ associate image object identifiers with *Mostly* and *Few* signal concept types.

Let us note that and_si and or_si are specialized relations of b_si. Also, am_si, al_si, mostly_si, few_si are specialized relations of q_si.

Image objects are represented by *Io* concepts and signal index and query structures are organized within a lattice of concept types. An instance of the signal facet is represented by a set of canonical CGs, each one containing an *Io* type possibly linked through signal conceptual relations to signal concept types.

There are two types of basic graphs controlling the generation of all signal facet graphs. The firsts are **index signal graphs**: [Io]→(b_si)→[B_{SI}]; [Io]→(q_si)→[Q_{SI}]. The seconds are **query signal graphs**: [Io]→(and_si)→[B_{And}]; [Io]→(or_si)→[B_{Or}] ; [Io]→(no_si)→[B_{No}] ; [Io]→(am_si)→[Q_{AM}]; [Io]→(al_si)→[Q_{AL}]; [Io]→(mostly_si)→[Q_{Mostly}] and [Io]→(few_si)→ [Q_{Few}].

The following index graphs are the representation of the signal facet in fig. 1:



5 The Querying Module

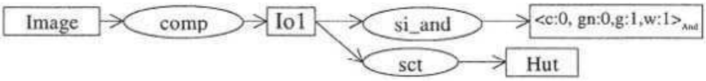
In image retrieval systems, the typical mode of user interaction relies on the query by example process [18]: the user provides a set of example images as an input, the system generates a query and then outputs images that are the most similar. This mode of interaction suffers from the fact that the user's need remains implicit, i.e. given the input images chosen by the user, the system has thus to use its knowledge of the image content to extract implicit information and construct a query. This process can be very complex and lead to ambiguities and poor retrieval performances when dealing with high-level characterizations of an image. Our conceptual architecture is based on a unified textual-based framework allowing a user to query over both the visual semantics and the signal facets. This obviously enhances user interaction since contrarily to query by example systems, the user becomes in 'charge' of the query process by making his needs explicit to the system through full textual querying. We will present in the following some queries involving boolean and quantified signal concepts, study their transcription within our conceptual framework and then deal with operations related to query processing. We will thus specify the organization of concept type lattices.

5.1 Query Expression

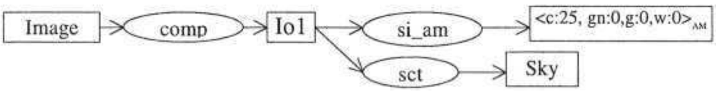
A general query is defined through a combination of selection criteria over the visual semantics and the signal facets. A query image q is represented by a 3-tuple (I_{IO}, q_{vs}, q_{si}) where I_{IO} is the set of image objects, q_{vs} and q_{si} are instances of the visual semantics and query signal facet models.

We propose to study the transcription in our model and then the processing of two types of queries for obvious space restrictions: the first one associates visual semantics with a boolean signal concept type (*And* signal concept type), the second associates visual semantics with a quantified signal concept type (*At Most* signal concept type).

5.1.1 Find images with a grey and white hut. In our formalism, it is translated as: $q=(I_{IO},q_{vs},q_{si})$ with $I_{IO}=\{Io1\}$; $q_{vs}=(\{Hut\},\{(Io1,Hut)\})$; $q_{si}=(\{<c{:}0,gn{:}0,g{:}1,w{:}1>_{And}\}, \{(Io1,<c{:}0,gn{:}0,g{:}1,w{:}1>_{And})\})$. In the CG representation formalism, we have:



5.1.2 Find images with a cloudy sky (At Most 25% of cyan). The transcription of this query in our conceptual framework is: $q=(I_{IO}, q_{vs}, q_{si})$: $I_{IO}=\{Io1\}$; $q_{vs}=(\{Sky\}, \{(Io1,Sky)\})$; $q_{si}=(\{<c{:}25,gn{:}0,g{:}0,w{:}0>_{AM}\}, \{(Io1,<c{:}25,gn{:}0,g{:}0,w{:}0>_{AM})\})$.
The transcription of this query in the CG representation formalism gives:



5.2 The Projection Operator

An operational model of image retrieval based on the CG formalism uses the graph projection operation for the comparison of a query graph and a document graph. This operator allows to identify within a graph g_1 sub-graphs with the same structure as a given graph g_2, with nodes being possibly restricted, i.e. their types are specialization of g_2 node types. If it exists a projection of a query graph Q within a document graph D then the document indexed by D is relevant for the query Q.

Formally, the projection operation $\wp : q \rightarrow d$ exists if there is a sub-graph of d verifying the two following properties:

– There is a unique document concept which is a specific of a query concept, this being valid for any query concept. This property ensures that all elements describing the query are present within the image document, and their image is unique.

– For any relation linking concepts c_{q1} and c_{q2} of q, there is the same relation between the two concepts c_{d1} and c_{d2} of d, such as $\wp(c_{q1}) = c_{d1}$ and $\wp(c_{q2}) = c_{d2}$.

At the implementation level, brute-force coding of the projection operation would result in exponential execution times. Based on the work in [19], we enforce the scalability of our framework using an adaptation of the inverted file approach for image retrieval. This technique consists in associating indexed keywords to the set of documents whose index contain it. Treatments that are part of the projection operation are performed during indexing following a specific organization of CGs which does not affect the expressiveness of the formalism.

5.3 Organizing Concept Type Lattices for Effective and Computationally Efficient Retrieval

In the following concept type lattices (fig. 4,5), the graphical arrow corresponds to a specialization operation and we consider that Cat={cyan, green, grey, white}.

5.3.1 Processing an *And* query. BSICs are organized within the *And* lattice (fig. 4) to process an *And* query. When a query such as "Find images with a grey and white hut" is formulated, it is first translated in a query CG with the semantic concept type *hut* processed by the lattice of semantic concept types (fig. 3) and the ASC $<c:0,gn:0,g:1,w:1>_{And}$. This ASC is then related to its equivalent BSIC as highlighted in fig. 4. The most relevant images provided by the system have a hut with grey and white only, this symbolic color distribution is represented by the highlighted BSIC (b_1) in fig. 4. Other images are composed of a hut with a color distribution including grey and white and at least one secondary color category. In the lattice, BSICs representing such color distributions are sons of b_1.

The general organization of this lattice is such that BSICs with a unique non-zero component are sons of the maximum virtual element T_{And}. They represent the perception of a unique color category in an image object. The BSIC with all non-zero components is at the bottom of the hierarchy, it is the minimum element noted $\perp_{And}$. This concept is a specialized concept of all BSICs presenting at least a non-zero component. Formally, we define a partial order in the *And* lattice of BSICs noted $\leq_{And}$ by:

$$\forall\ a,b \in B_{SI}\ \ a \leq_{And} b \Leftrightarrow [a = \bot_{And} \vee b = \top_{And}] \vee [\neg\exists k \in [1,C_{cat}] \,/\, b_{[k]} = 1 \wedge a_{[k]} = 0] \qquad (1)$$

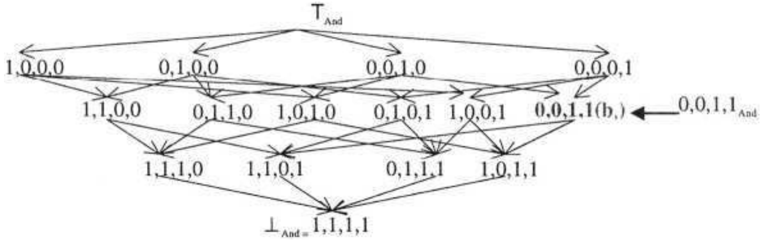


Fig. 4. Lattice processing *And* queries

5.3.2 Processing an *At Most* query. When a query such as "Find images with a cloudy sky (i.e. with a color distribution that includes at most 25% of cyan)" is formulated, it is translated in a query CG with the semantic concept type *sky* processed by the lattice of semantic concept types (fig. 3) and the AMSC $<c:25,g:0,gn:0,w:0>_{AM}$. However, the link between this AMSC and its equivalent QSIC is not straightforward. Therefore we introduce a new category of concepts types bridging the gap between AMSCs and QSICs by taking into account dominant color categories (i.e. categories mentioned in a query as they have a higher importance in the ordering process of signal concepts within the lattice, other color categories are called secondary). These concept types are QSICs with dominant d_{AM}, where d_{AM} is the set of dominant color categories. They are supported by a vector structure $v_{AMd}[i]$ with a number of elements equal to $C_{cat}+1$. The $v_{AMd}[i]_{i \in [1,Ccat+1]}$ values such that $C_i \in d_{AM}$ are the maximum pixel percentages of dominant color categories and the $v_{AMd}[j]_{j \in [1,Ccat+1]}$ such that $j \neq i$ correspond to the pixel percentages of secondary color categories ranked in ascending order. A component summing pixel percentages of secondary color categories noted Σ is introduced. By construction, this element is the maximum value among the $v_{AMd}[j]_{j \in [1,Ccat+1]}$. QSICs with dominant d_{AM} are therefore specializations of AMSCs and generalizations of QSICs and link AMSCs to QSICs. The AMSC $<c:25,g:0,gn:0,w:0>_{AM}$ is related to its equivalent QSIC with dominant {cyan}: $<\underline{25},25,25,25,75>$ as highlighted in fig. 5a. As a matter of fact, the most relevant images provided by the system have a sky with 25% of cyan and a remaining proportion uniformly distributed between the 3 secondary color categories (25% each in our example). Others are images with a sky having a color distribution that includes less than 25% of cyan, the remaining proportion *p* being in the best cases uniformly distributed between the 3 secondary color categories.

Formally, sub-lattices of AMSCs with dominant d_{AM} (framed structure in fig. 5a) are partially ordered by $\leq_{AM}$:

$$\forall\ a,b \text{ QSICs with dominant } d_{AM},\ a \leq_{AM} b \Leftrightarrow [a=\bot_{AM} \vee b=\top_{AM}] \vee [\forall j \in [1,C_{Cat}] \,/\, Cat_j \in d_{AM},\ 1 \leq a_{[j]} \leq b_{[j]}] \qquad (2)$$

Sub-lattices of concept types with components corresponding to dominant color categories being equal (framed structure in fig. 5b) are partially ordered by $\leq_{AM\_eq}$:

$$\forall \text{ a,b QSICs with dominant } d_{AM} \text{ having components that correspond to dominant color categories being equal, } a \leq_{AM\_eq} b \Leftrightarrow (\forall\, j, k \in [1, C_{Cat} + 1] \,/\, Cat_j \notin d_{AM} \wedge Cat_k \notin d_{AM}, \Sigma_{j,k} | b_{[j]} - b_{[k]} | \leq \Sigma_{j,k} | a_{[j]} - a_{[k]} |) \quad (3)$$

Let us note than the *At Least* lattice has a symmetric organization and will not be dealt with for space restriction.

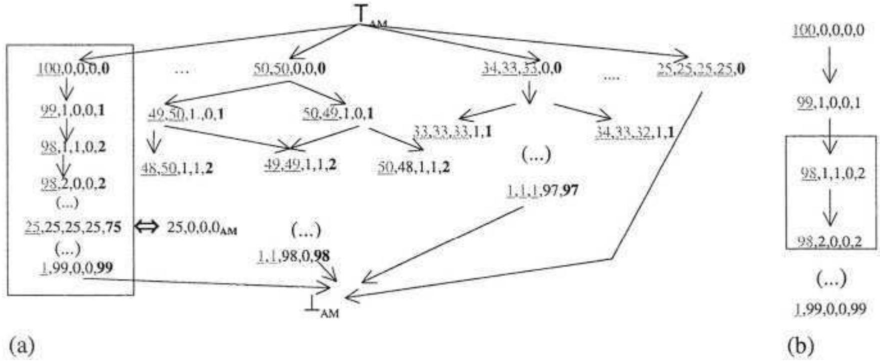


Fig. 5. (a) Sub-lattice of At Most signal concepts, (b) with dominant {C1=cyan}

5.3.3 Processing a query with a literal quantifier. *Mostly* and *Few* queries involving literal quantifiers, e.g. "Find images with a bright sky (mostly cyan)" or "Find images with a cloudy sky (few cyan)", are processed accordingly to *At Most* and *At Least* queries. Indeed, the quantifier Mostly corresponds to the numeral quantification 'At Least 50%' and the quantifier Few is linked to the numeral quantification 'At most 10%'. As a matter of fact, processing these queries will not affect the computational efficiency of our model as it is based on AMSCs and ALSCs concept type lattices.

6 Experimental Results

We have presented a conceptual architecture in which semantic and signal features are integrated to achieve higher expressivity as far as querying is concerned and increased retrieval accuracy. We will describe here the SIAIR image retrieval system that is an implementation of the theoretical framework presented and present several experimentation results.

The SIAIR image retrieval system implements the formal framework presented in this paper, the supported mode of interaction relying on keyword-based search. When a user enters a query, it is translated in a CG query graph as developed in section 5. It is then processed and images given by the system are ranked and displayed according to their relevance with respect to the query.

Validation experiments are carried out on a corpus of 2500 personal color photographs collected over a period of five years and used as a validation corpus in world-class publications [9,15] (fig. 1 displays a typical photograph which belongs to this collection). Dealing with personal photographs instead of professional collections (e.g. Corel images) is guided by our research problematic which is the specification of an expressing framework enhancing techniques that allow a user to index and query over a collection of home photographs. Moreover, the quality of home photographs is not as good as the quality of professional images which leads to retrieval results being generally poorer than those for the Corel images.

Image objects within the 2500 photographs are automatically assigned a semantic concept as presented in section 3 and are characterized with conceptual signal structures presented in section 4. Eleven color categories (red, green, blue, yellow, cyan, purple, black, skin, white, grey, orange) empirically spotlighted in [4] are described in the HVC perceptually uniform space by a union of brightness, tonality and saturation intervals [13].

Given an image corpus, we wish to retrieve photographs that represent elaborate scenes involving signal characterization. We specify 22 image scenes (e.g. night, swimming-pool water...) and select within the corpus for each scene all images which are relevant. The evaluation of our formalism is based on the notion of **image relevance** which consists in quantifying the correspondence between index and query images.

We compare our approach with both state-of-the-art signal and semantics-based approaches, namely “HSV local” and “Visual keywords”. The HSV local method is based on the specification of ten key colors (red, green, blue, black, grey, white, orange, yellow, brown, pink) in the HSV color space adopted by the original PicHunter system [3]. The similarity matching between two images is computed as the weighted average of the similarities between corresponding blocks of the images. As a matter of fact, this method is equivalent to locally weighted color histograms.

Visual keywords [8,9,15] are intuitive and flexible visual prototypes extracted or learned from a visual content domain with relevant semantic labels. A set of 26 specified visual keywords are learned using a neural network, with low-level features computed for each training region as an input for this network. An image is indexed to multi-scale, view-based recognition against these 26 visual keywords, recognition results across multiple resolutions are aggregated according to spatial tessellation. It is then represented by a set of local visual keyword histograms with each bin corresponding to the aggregation of recognition results. The similarity matching between two images is defined as the weighted average of the similarities between their corresponding local visual keywords histograms. The HSV local and Visual Keywords methods are presented here to compare the results of usual signal-based and semantic-based approaches to our framework combining both of these approaches.

For each of the 22 image scene descriptions (e.g. swimming-pool water), we construct relevant textual query terms using corresponding semantic and signal concepts as input to the SIAIR system (e.g. “Find images with mostly cyan” for swimming-pool water). Also each image scene description is translated in textual signal data as input to the HSV local approach (“Find images with cyan” for swimming-pool water) and in relevant visual keywords to be processed by the Visual keywords system (“Find images with a sky” for swimming-pool water). Curves associated with the *Q_SymbColor, Q_Symb* and *Q_Color* legends (fig. 6) illustrate respectively the results in recall and precision obtained by SIAIR, the Visual Keywords and the HSV local systems.

The average precision of SIAIR (0.5854) is approximately five times higher than the average precision of the Visual Keywords method (0.1115) and approximately 3,5 times higher than the value of average precision of the HSV local method (0.168). We notice that improvements of the precision values are significant at all recall values. This shows that when dealing with elaborate queries which combine multiple sources of information (here visual semantics and signal features) and thus require a higher level of abstraction, the use of an "intelligent" and expressive representation formalism (here the CG formalism within our framework) is crucial. As a matter of fact, SIAIR complements both state-of-the-art signal-based approaches by proposing a framework for semantic characterization and state-of-the-art semantic-based methods through signal conceptual integration, which enriches indexing languages and expands usual querying frameworks restricted to a reduced set of extracted or learned keywords (in this case the visual keywords).

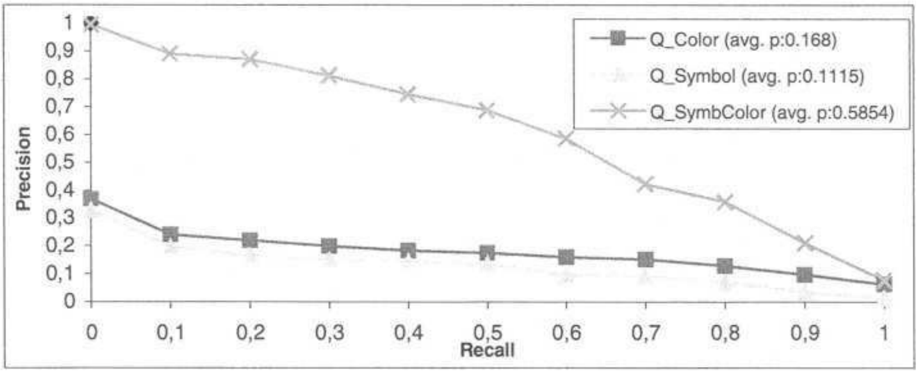


Fig. 6. Recall/Precision curves

7 Conclusion

We have proposed within the scope of this paper the formal specification of a framework combining the two existing approaches in image retrieval, i.e. signal and symbolic within a strongly-coupled architecture to achieve greater retrieval accuracy. It is instantiated by an operational model based on the CG formalism, which allows to define an image representation and a correspondence function to compare index document and query graphs. Our work has contributed both theoretically and at the experimental level to the image retrieval research topic. We have specified image objects, abstract structures representing visual entities within an image in order to operate image indexing and retrieval operations at a higher level of abstraction than state-of-the-art frameworks. We have formally described the visual semantics and the signal facets that define the conceptual information conveyed by image objects and have finally proposed a unified and rich framework for querying over both visual semantics and signal data. At the experimental level, we have implemented and evaluated our framework. The results obtained allowed us to validate our approach and stress the relevance of the signal/semantics integration.

References

1. Berlin, B. & Kay, P.: Basic Color Terms: Their universality and Evolution. UC Press (1991)
2. Carson, C. & al.: Blobworld: A System for Region-Based Image Indexing and Retrieval. VISUAL (1999) 509-516
3. Cox, I. & al.: The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments. IEEE Trans. Image Processing, vol.9, no.1 (2000) 20-37
4. Gong, Y. & Chuan, H. & Xiaoyi, G.: Image Indexing and Retrieval Based on Color Histograms. Multimedia Tools and Applications II (1996) 133-156
5. Gupta, A. & Weymouth, T. & and Jain, R: Semantic queries with pictures: The VIMSYS model. VLDB (1991) 69-79
6. La Cascia & al.: Combining Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web. IEEE Workshop on CB Access of Im. and Vid. Lib. (1998) 24-28
7. Lammens, J.M.: A computational model of color perception and color naming. PhD, State Univ. of New York, Buffalo (1994)
8. Lim, J.H.: Explicit query formulation with visual keywords. ACM MM (2000) 407-412
9. Lim, J.H. & al.: Home Photo Content Modeling for Personalized Event-Based Retrieval. IEEE Multimedia, vol.10, no.4 (2003)
10. Lu, Y. & al.: A unified framework for semantics and feature based relevance feedback in image retrieval systems. ACM MM (2000) 31-37
11. Ma, W.Y. & Manjunath, B.S. 1997.: NeTra: A toolbox for navigating large image databases. ICIC (1997) 568-571
12. Mechkour, M.: **EMIR^2**: An Extended Model for Image Representation and Retrieval. DEXA (1995) 395-404
13. Miyahara, M. & Yasuhiro Yoshida, Y.: Mathematical Transform of (R,G,B) Color Data to Munsell (H,V,C) Color Data. SPIE Vol. 1001 (1988) 650-657
14. Mojsilovic, A. & Rogowitz, B.: Capturing image semantics with low-level descriptors. ICIP (2001) 18-21
15. Mulhem, P. & Lim, J.H.: Symbolic photograph content-based retrieval. ACM CIKM (2002) 94-101
16. Niblack, W. et al.: The QBIC project : Querying images by content using color, texture and shape. SPIE, Storage and Retrieval for Image and Video Databases (1993) 40-48
17. Ounis, I. & Pasca, M.: RELIEF: Combining expressiveness and rapidity into a single system. ACM SIGIR (1998) 266-274
18. Smeulders, A.W.M. & al.: Content-based image retrieval at the end of the early years. IEEE PAMI, 22(12) (2000) 1349-1380
19. Smith, J.R. & Chang, S.F.: VisualSEEk: A fully automated content-based image query system. ACM MM (1996) 87-98
20. Sowa, J.F. “Conceptual structures : information processing in mind and machine”. Addison-Wesley publishing company (1984)
21. Zhou, X.S. & Huang, T.S.: Unifying Keywords and Visual Contents in Image Retrieval. IEEE Multimedia 9(2) (2002) 23-33

Improving Retrieval Effectiveness by Reranking Documents Based on Controlled Vocabulary

Jaap Kamps

Language & Inference Technology Group
ILLC, University of Amsterdam
http://lit.science.uva.nl/

Abstract. There is a common availability of classification terms in online text collections and digital libraries, such as manually assigned keywords or key-phrases from a controlled vocabulary in scientific collections. Our goal is to explore the use of additional classification information for improving retrieval effectiveness. Earlier research explored the effect of adding classification terms to user queries, leading to little or no improvement. We explore a new feedback technique that reranks the set of initially retrieved documents based on the controlled vocabulary terms assigned to the documents. Since we do not want to rely on the availability of special dictionaries or thesauri, we compute the meaning of controlled vocabulary terms based on their occurrence in the collection. Our reranking strategy significantly improves retrieval effectiveness in domain-specific collections. Experimental evaluation is done on the German GIRT and French Amaryllis collections, using the test-suite of the Cross-Language Evaluation Forum (CLEF).

1 Introduction

Online text collections and digital libraries commonly provide additional classification information, such as controlled vocabulary terms in scientific collections. These classifications can be assigned either manually, or automatically [1]. The widespread use of additional classification terms prompts the question whether this additional information can be used to improve retrieval effectiveness. That is, when considering retrieval queries that do not use classification terms, can we make use of the fact that the retrieved documents have classification terms assigned to them? In IR parlance, this is a form of feedback.

Feedback or query expansion methods have a long history in information retrieval. This dates back, at least, to the studies of Sparck Jones [2,3] in which the collection is analyzed to provide a similarity thesaurus of word relationships. This type of approach is called *global* feedback in [4], which introduces a *local* feedback variant in which the initially retrieved documents are analyzed. There is mixed evidence on the effectiveness of global feedback. Local feedback methods are generally more effective, and the combination, by using global analysis techniques on the local document set, tends to be most effective [5].

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 283–295, 2004.

An obvious feedback approach to exploiting classification information is to expand the original queries with (some of) the classification terms. This has received a fair amount of attention, especially in the medical domain where excellent resources exist. Srinivasan [6] investigates automatic query expansion with MeSH terms using the MEDLINE collection, based on a statistical thesaurus. Her finding is that query expansion with controlled vocabulary terms leads to improvement, but the effect is overshadowed by standard blind feedback. Hersh et al. [7] investigate various ways of expanding medical queries with UMLS Metathesaurus terms, and find a significant drop in retrieval effectiveness. Recently, French et al. [8] showed that a significant improvement of retrieval effectiveness is possible for query expansion with MeSH terms. However, they select the terms to be added by analyzing the set of human-judged, relevant documents. This gold standard experiment does not solve the problem of how to select the appropriate controlled vocabulary terms in the absence of full relevance information. Gey and Jiang [9] found a mild improvement of retrieval effectiveness when GIRT queries were expanded using thesaurus terms.

In sum, there is no equivocal evidence that fully automatically expanding queries with controlled vocabulary terms from initially retrieved documents leads to significant improvement of retrieval effectiveness. This motivated us to experiment with an alternative to expanding queries with classification terms. We explored a new feedback technique that reranks the set of initially retrieved documents based on the controlled vocabulary terms assigned to the documents. We use essentially a combination of global and local feedback techniques. On the one hand, we use a global feedback technique to analyze the usage of controlled vocabulary in the collections. The rationale for this is that we do not want to rely on the availability of special dictionaries or thesauri. Our approach is similar to latent semantic indexing [10]. We estimate the similarity of controlled vocabulary terms from their usage in the collections. Next, we apply dimensional reduction techniques, resulting in a low dimensional controlled vocabulary space. On the other hand, we use a local feedback technique for reranking the set of initially retrieved documents. Our strategy is to rerank the set of initially retrieved documents by their distance (based on the assigned controlled vocabulary terms) to the top-ranked retrieved documents.

The rest of this paper is structured as follows. Next, in section 2, we investigate the controlled vocabulary usage in scientific collections, and show how a similarity or distance measure can be used to obtain similarity vectors for the controlled vocabulary terms. Then, in section 3, we will provide some details of the experimental setup, and propose two document reranking strategies. In section 4, we investigate the results of the two reranking strategies, and their impact on retrieval effectiveness. Finally, in section 5, we discuss our results and draw some conclusions.

2 Controlled Vocabulary

The cross-language evaluation forum (CLEF [11]) addresses four different cross-lingual information retrieval tasks: monolingual, bilingual, multilingual, and

Table 1. Statistics about the GIRT and Amaryllis collections (stopwords are not included)

Collection	GIRT	Amaryllis
Documents	76,128	148,688
Size (Mb)	151	196
Words per Document	700	735
Queries	24	25
Words per Query	9.28	36.04
Relevant Documents per Query	40.0	80.7

Table 2. GIRT topic 051 (only title and description fields)

⟨DE-title⟩ *Selbstbewusstsein von Mädchen*	⟨EN-title⟩ Self-confidence of girls
⟨DE-desc⟩ *Finde Dokumente, die über den Verlust des Selbstbewusstseins junger Mädchen während der Pubertät berichten.*	⟨EN-desc⟩ Find documents which report on the loss of self-confidence of young girls during the puberty.

Table 3. Amaryllis topic 001 (only title and description fields)

⟨FR-title⟩ *Impact sur l'environnement des moteurs diesel*	⟨EN-title⟩ The impact of diesel engine on environment
⟨FR-desc⟩ *Pollution de l'air par des gaz d'échappement des moteurs diesel et méthodes de lutte antipollution. Emissions polluantes (NOX, SO2, CO, CO2, imbrûlés, ...) et méthodes de lutte antipollution*	⟨EN-desc⟩ Air pollution by the exhaust of gas from diesel engines and methods of controlling air pollution. Pollutant emissions (NOX, SO2, CO, CO2, unburned product, ...) and air pollution control

domain-specific retrieval. For domain-specific retrieval at CLEF, the scientific collections of GIRT (German) and Amaryllis (French) are used. Table 1 gives some statistics about the test collections. Notice that the Amaryllis queries are long, due to the use of multi-sentence descriptions. Table 2 show one of the GIRT topics, and Table 3 shows one of the Amaryllis topics.

The GIRT collection contains (abstracts of) documents from German social science literature published between 1978 and 1996 [12]. The documents are also classified by controlled vocabulary terms assigned by human indexers, using the controlled-vocabulary thesaurus maintained by GESIS [13]. The average number of controlled vocabulary terms in a document is 9.91. Table 4 gives some of the characteristics of controlled vocabulary in the GIRT and the Amaryllis collections. The Amaryllis collection contains (abstracts of) documents in French from various scientific fields. The average number of manually assigned controlled vocabulary terms in a document is 10.75.

We want to compute the similarity of controlled vocabulary terms based on their occurrence in the collection. Our working hypothesis is that controlled vo-

Table 4. Controlled Vocabulary Usage in the GIRT and Amaryllis collections

	GIRT		Amaryllis	
Used terms	6,745		125,360	
Occurrences	755,333	(704 doubles)	1,599,653	(562 doubles)
Most frequent	29,561	*Bundesrepublik Deutschland*	20,514	*Homme*
	9,246	*Frau*	17,283	*France*
	6,133	*historische Entwicklung*	7,888	*Traitement*
	4,736	*Entwicklung*	6,619	*Etude expérimentale*
	4,451	*neue Bundesländer*	5,987	*Etude cas*
	3,645	*DDR*	4,319	*Diagnostic*
	3,445	*Österreich*	4,179	*Modélisation*
	3,341	*Entwicklungsland*	4,171	*Enfant*
	3,025	*Betrieb*	4,130	*Etude comparative*
	3,012	*geschlechtsspezifische Faktoren*	3,954	*Article synthèse*

cabulary terms that are frequently assigned to the same documents, will have similar meaning. We only give an outline of the used approach here, since we apply well-known techniques. For the convenience of interested readers, a detailed description is provided in Appendix A. We determine the number of occurrences of controlled vocabulary terms and of co-occurrences of pairs of controlled vocabulary terms use in the collection, and use these to define a distance metric over the controlled vocabulary terms. Specifically, we use the Jaccard similarity coefficient on the log of (co)occurrences, and use 1 minus the Jaccard score as a distance metric [14]. For creating manageable size vectors for each of the controlled vocabulary terms, we reduce the matrix using metric multi-dimensional scaling techniques [15]. For all calculations we used the best approximation of the distance matrix on 100 dimensions. This results in a 100-dimensional vector for each of the 6,745 controlled vocabulary terms occurring in the GIRT collection. The Amaryllis collection uses a much richer set of 125,360 controlled vocabulary terms. We select only the controlled vocabulary terms occurring at least 25 times in the collection. Thus, we end up with a 100-dimensional vector for the 10,274 most frequent controlled vocabulary terms in the Amaryllis collection. Basically, we now have a vector space for the controlled vocabulary terms, where related terms will be at a relatively short distance, and unrelated terms far apart.

Note that we only have vectors for the controlled vocabulary terms. However, there are straightforward ways to map documents and topics into the controlled vocabulary space. For each document we collect the assigned controlled vocabulary terms from the collection. We have a vector in the controlled vocabulary space for each of the controlled vocabulary terms. We define the vector for the document to be simply the mean score for each of the controlled vocabulary term vectors. We can also create vectors for topics, based on which documents are retrieved by an information retrieval system (here, we use the 10 best ranked documents). For each topic we consider the top-ranked documents, and define the vector for the topic to be the weighted mean score of the document vectors. We give each document a weight corresponding to its retrieval status value (RSV).

3 Experimental Setup

All retrieval experiments were carried out with the FlexIR system developed at the University of Amsterdam [16]. The main goal underlying FlexIR's design is to facilitate flexible experimentation with a wide variety of retrieval components and techniques. FlexIR is implemented in Perl and supports many types of preprocessing, scoring, indexing, and retrieval tools. One of the retrieval models underlying FlexIR is the standard vector space model. All our runs use the `Lnu.ltc` weighting scheme [17] to compute the similarity between a query and a document. For the experiments on which we report in this paper, we fixed the slope at 0.2; the pivot was set to the average number of unique words per document.

For the GIRT and Amaryllis collections, we index both the free-text of the documents, i.e., the title and body or abstract of articles, as well as the manually assigned controlled vocabulary terms. Our index contains the words as they occur in the collection with only limited sanitizing, i.e., we remove punctuation; apply case-folding; map marked characters to the unmarked tokens; and remove stopwords. We employ generic lists with stopwords, with 155 stopwords for French, and 231 stopwords for German. We do not apply further morphological normalization; see [18] for an overview of the effectiveness of stemming and n-gramming for monolingual retrieval in German and French.

From the indexes we obtain a baseline run per collection. For our reranking experiments, we use the controlled vocabulary space to rerank the documents initially retrieved in the baseline run. For all the retrieved documents, we extract the assigned controlled vocabulary terms from the collection. Then, we calculate document vectors for all the documents, by calculating the mean of the vectors for controlled vocabulary terms assigned to them. Finally, we calculate a vector for the topic, by calculating the weighted mean of the 10 top-ranked documents. Based on the topic and document vectors, we experiment with two reranking feedback strategies.

Naive reranking. We have a vector for each of the topics, and for each of the retrieved documents. Thus, ignoring the RSV of the retrieved documents, we can simply rerank all documents by increasing euclidean distance between the document and topic vectors. Since RSVs should be decreasing by rank, we use 1 minus the distance as the new RSV.

Combined reranking. We investigate a more conservative reranking by combining the two sources of evidence available: the original text-based similarity scores of the baseline run, and the controlled vocabulary-based distances which are calculated as in the naive reranking. The scores were combined in the following manner. Following Lee [19], both scores are normalized using $RSV_i' = \frac{RSV_i - min_i}{max_i - min_i}$. We assigned new weights to the documents using the summation function used by Fox & Shaw [20]: $RSV_{new} = RSV_1' + RSV_2'$. This combination results in a less radical reranking of documents.

Since we are interested in the interaction of our reranking feedback with standard blind feedback, we do three sets of experiments. In the first set we evaluate our reranking feedback using the original queries. In the second set of experiments, we apply standard blind feedback to expand the original queries with related terms from the free-text of the documents. Term weights were recomputed using the standard Rocchio method [21], where we considered the top 10 documents to be relevant and the bottom 500 documents to be non-relevant. We allowed at most 20 terms to be added to the original query. In the third set of experiments, we investigate the effectiveness of the reranking feedback using the expanded queries from the second set of experiments.

Finally, to determine whether the observed differences between two retrieval approaches are statistically significant, we used the bootstrap method, a non-parametric inference test [22,23]. The method has previously been applied to retrieval evaluation by, e.g., Wilbur [24] and Savoy [25]. We take 100,000 resamples, and look for significant improvements (one-tailed) at significance levels of 0.95 (*), 0.99 (**) and 0.999 (***).

4 Experimental Results

4.1 Reranking Feedback

In the first set of experiments, we study the effectiveness of our new reranking feedback method using the original queries. We create baseline runs using the indexes of the free-text and controlled vocabulary terms of the documents. We use the title and description fields of the CLEF 2002 topics. The results are shown in Table 5. The resulting baseline run for GIRT has a mean average precision (MAP) of 0.2063. The resulting baseline run for Amaryllis has a MAP of 0.2778. Next, we employ the naive reranking strategy to the respective baseline runs (as described in Section 3). The result of the naive reranking is negative: we find a decrease in performance for both GIRT and Amaryllis. The drop in performance for Amaryllis is even significant. Does this mean that the calculated topic vector is not adequately representing the content of the topics? Or is it a result of our radical reranking approach?

We investigate this by employing the combined rerank strategy that takes both the text-based similarity score, as well as the distance to the topic vector into account (as described in Section 3). The results of the combined rerank runs are also shown in Table 5: for GIRT the MAP improves to 0.2487, and

Table 5. Mean average precision scores for the baseline runs, the naive rerank runs, and the combined rerank runs, using CLEF 2002 topics. Best scores are in boldface, significance $^{*} = p < .05$, $^{**} = p < .01$, $^{***} = p < .001$

	GIRT		Amaryllis	
Run	MAP	% Change	MAP	% Change
Baseline	0.2063		0.2778	
Naive rerank	0.1973	-4.4%	0.1829	-34.2%***
Combined rerank	**0.2487**	+20.6%***	**0.3059**	+10.1%**

for Amaryllis the MAP improves to 0.3059. The respective improvements are +20.6% (GIRT) and +10.1% (Amaryllis). The improvement of both combined rerank runs is statistically significant. Thus we find evidence that the combined rerank strategy is significantly improving retrieval effectiveness.

4.2 Rocchio Blind Feedback

In a second set of experiments, we study the effectiveness of standard blind feedback. A possible explanation of the observed improvement due to reranking feedback is that it functions roughly like standard blind feedback. The results of applying Rocchio blind feedback (as described in Section 3) are shown in Table 6. We see that Rocchio blind feedback is promoting retrieval effectiveness. The resulting blind feedback runs for GIRT have a MAP of 0.2209 (an improvement of +7.1% over the unexpanded queries). The resulting blind feedback runs for Amaryllis have a MAP of 0.2986 (an improvement of +7.5%). Blind feedback is improving retrieval effectiveness, although the improvements are not significant. Note also that improvement due to blind feedback is less than the improvement due to combined reranking as discussed in our first set of experiments. Thus, when comparing the relative effectiveness of both types of feedback, the reranking feedback meets and exceeds the effectiveness of standard Rocchio blind feedback.

Table 6. Mean average precision scores for the baseline runs and the Rocchio blind feedback runs using CLEF 2002 topics. Best scores are in boldface, significance $^{\star} = p < .05$, $^{\star\star} = p < .01$, $^{\star\star\star} = p < .001$

	GIRT		Amaryllis	
Run	MAP	% Change	MAP	% Change
Baseline	0.2063		0.2778	
Blind feedback	**0.2209**	+7.1%	**0.2986**	+7.5%

4.3 Rocchio Blind Feedback Plus Reranking Feedback

In the third set of experiments, we investigate whether the improvement of retrieval effectiveness we found in the first set of experiments is supplementary to the effects of standard blind feedback we found in the second set of experiments. That is, the difference with the first set of experiments is that we now use queries that have been expanded by Rocchio blind feedback. The results are shown in Table 7, note that we now compare the improvement relative to the expanded queries, and not relative to the earlier baseline run. The results of the naive reranking strategy are no better than in the first set of experiments: both runs show a drop in performance, and the decrease in performance is significant for Amaryllis. The combined rerank strategy turns out to be effective again. For GIRT the MAP is 0.2481 (+12.3% over the expanded queries) and for Amaryllis

Table 7. Mean average precision scores for the Rocchio blind feedback runs, the naive rerank runs, and the combined rerank runs, using CLEF 2002 topics. Best scores are in boldface, significance $^\star = p < .05$, $^{\star\star} = p < .01$, $^{\star\star\star} = p < .001$

	GIRT		Amaryllis	
Run	MAP	% Change	MAP	% Change
Blind feedback	0.2209		0.2986	
Naive rerank	0.1831	-17.1%	0.2025	-32.2%***
Combined rerank	**0.2481**	+12.3%*	**0.3197**	+7.1%*

the MAP is 0.3197 (+7.1%). Both improvements are statistically significant. So, we find evidence that the combined rerank strategy is significantly improving retrieval effectiveness, on top of the effect due to blind feedback.

When comparing the combined reranking scores with those obtained in the first set of experiments, we notice the following. The combined reranking score for Amaryllis expanded queries is 4.5% higher than the score for the unexpanded queries. However, the combined reranking score for the expanded GIRT queries is 0.2% lower than the score for the unexpanded queries. Thus in this case, the use Rocchio blind feedback is hurting the score, possibly due to topic drift influencing the retrieved top 10 documents for some of the topics.[1]

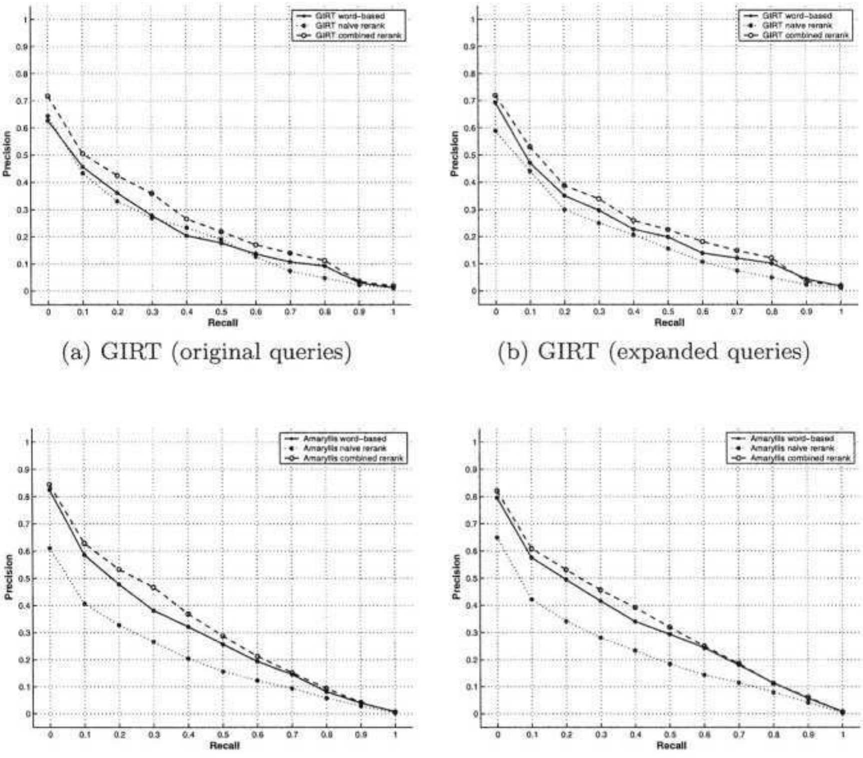
Figure 1 plots the recall-precision curves for the reranking feedback experiments we conducted. We have shown that the combined reranking strategy leads to a significant improvement of retrieval effectiveness. This also shows that the topic vector can be used to capture the content of the topic. In turn, this demonstrates the viability of our approach to derive the meaning of controlled vocabulary terms from their occurrence in the collection.

5 Discussion and Conclusions

This paper introduced a new feedback technique that reranks the set of initially retrieved documents based on the controlled vocabulary terms assigned to the documents. Our reranking strategy significantly improved retrieval effectiveness in domain-specific collections, above and beyond the use of standard Rocchio blind feedback.

Our method is specifically tailored for collections that provide additional classification of documents, such as manually assigned controlled vocabulary terms in scientific collections. We derived a controlled vocabulary thesaurus based on their (co)occurrences in the collections. Similar approaches have been proposed since the advent of information retrieval. For example, Sparck Jones [3] discusses the clustering of words based on their co-occurrence. The dimensional reduction techniques we used are similar to those used in latent semantic indexing [10]. Our focus was on the classication terms in the collection, although the same

[1] Recent evidence suggest that Rocchio feedback is promoting overall performance, but hurts performance on the poorly performing topics [26].



(a) GIRT (original queries)

(b) GIRT (expanded queries)

(c) Amaryllis (original queries)

(d) Amaryllis (expanded queries)

Fig. 1. Interpolated recall-precision averages for the naive rerank runs and the combined rerank runs on the original and expanded queries using CLEF 2002 topics

techniques can be applied to all, or a selection of, words in the collection. Gauch et al. [27,28] use a corpus analysis approach for query expansion. Schütze and Pedersen [29] use a cooccurrence-based thesaurus to derive context vectors for query words. Our approach differs from earlier work by its focus on the reranking of the initially retrieved document, based on the controlled vocabulary terms assigned to the documents. The queries only play a role in the retrieval of the initial set of documents. Perhaps closest in spirit is the work of Jin et al. [30], proposing a language model that takes classification labels into account.

Experimental evaluation was done on the German GIRT and French Amaryllis collections, using the test-suite of the Cross-Language Evaluation Forum [11]. We experimented with two reranking strategies. The first strategy, a naive ranking based solely on the distances, generally showed a drop in performance. The second strategy, a combined reranking using evidence from both the text-based relevance score and the controlled vocabulary-based distances, showed a significant improvement of retrieval effectiveness. To investigate how the improvement due to reranking relates to standard blind feedback, we conducted further experiments and showed that reranking feedback is more effective than Rocchio blind feedback. Moreover, we can apply reranking to expanded queries leading, again, to a significant improvement. For one of the collections, however, the score for combined reranking feedback is lower for the expanded queries than for the original queries. Thus, were in earlier research the gain due to query expansion with controlled vocabulary was overshadowed by the gain due to standard blind feedback, we here see that reranking feedback is overschadowing the gain due to Rocchio feedback.

There are obvious differences between standard blind feedback and reranking feedback, for example, an important effect of query expansion is the retrieval of additional relevant documents, i.e., an improvement of recall, whereas a reranking strategy can only improve the ranking of relevant documents, i.e., an improvement of precision. Further experimentation is needed to fully assess the relative impact of both feedback methods, and to uncover the underlying mechanisms responsible for their effectiveness. This should take into account similar results in interactive retrieval, where relevance feedback tends to produce more accurate results than query reformulation [31].

Acknowledgments. This research is supported by the Netherlands Organization for Scientific Research (NWO, grants 400-20-036 and 612.066.302). Many thanks to Maarten de Rijke and Börkur Sigurbjörnsson for comments and suggestions.

References

1. Lewis, D.D.: An evaluation of phrasal and clustered representations on a text categorization task. In Belkin, N., Ingwersen, P., Pejtersen, A.M., Fox, E., eds.: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, New York NY, USA (1992) 37–50
2. Sparck Jones, K., Needham, R.: Automatic term classification and retrieval. Information Processing & Management **4** (1968) 91–100
3. Sparck Jones, K.: Automatic Keyword Classification for Information Retrieval. Butterworth, London (1971)
4. Attar, R., Fraenkel, A.S.: Local feedback in full-text retrieval systems. Journal of the Association of Computing Machinery **24** (1977) 397–417
5. Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In Frei, H.P., Harman, D., Schaübie, P., Wilkinson, R., eds.: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, New York NY, USA (1996) 4–11
6. Srinivasan, P.: Query expansion and MEDLINE. Information Processing & Management **34** (1996) 431–443
7. Hersh, W., Price, S., Donohoe, L.: Assessing thesaurus-based query expansion using the UMLS metathesaurus. In: Proceedings of the 2000 AMIA Annual Fall Symposium. (2000) 344–348
8. French, J.C., Powell, A.L., Gey, F., Perelman, N.: Exploiting a controlled vocabulary to improve collection selection and retrieval effectiveness. In: Proceedings of the tenth International Conference on Information and Knowledge Management, ACM Press (2001) 199–206

9. Gey, F.C., Jiang, H.: English-German cross-language retrieval for the GIRT collection–exploiting a multilingual thesaurus. In: Proceedings of the Eighth Text REtrieval Conference (TREC-8), National Institute for Standards and Technology, Washington, DC (1999)
10. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407
11. CLEF: Cross language evaluation forum (2003) `http://www.clef-campaign.org/`.
12. Kluck, M., Gey, F.C.: The domain-specific task of CLEF - specific evaluation strategies in cross-language information retrieval. In Peters, C., ed.: Cross-Language Information Retrieval and Evaluation, CLEF 2000. Volume 2069 of Lecture Notes in Computer Science., Springer (2001) 48–56
13. Schott, H., ed.: Thesaurus Sozialwissenschaften. Informationszentrum Sozialwissenschaften, Bonn (2002) 2 Bände: Alphabetischer und systematischer Teil.
14. Gower, J.C., Legendre, P.: Metric and euclidean properties of dissimilarity coefficients. Journal of Classification **3** (1986) 5–48
15. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Chapman & Hall, London UK (1994)
16. Monz, C., de Rijke, M.: Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In Peters, C., Braschler, M., Gonzalo, J., Kluck, M., eds.: Evaluation of Cross-Language Information Retrieval Systems, CLEF 2001. Volume 2406 of Lecture Notes in Computer Science., Springer (2002) 262–277
17. Buckley, C., Singhal, A., Mitra, M.: New retrieval approaches using SMART: TREC 4. In Harman, D.K., ed.: The Fourth Text REtrieval Conference (TREC-4), National Institute for Standards and Technology. NIST Special Publication 500-236 (1996) 25–48
18. Hollink, V., Kamps, J., Monz, C., de Rijke, M.: Monolingual document retrieval for European languages. Information Retrieval **7** (2004) 33–52
19. Lee, J.H.: Combining multiple evidence from different properties of weighting schemes. In Fox, E.A., Ingwersen, P., Fidel, R., eds.: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, New York NY, USA (1995) 180–188
20. Fox, E.A., Shaw, J.A.: Combination of multiple searches. In Harman, D.K., ed.: The Second Text REtrieval Conference (TREC-2), National Institute for Standards and Technology. NIST Special Publication 500-215 (1994) 243–252
21. Rocchio, Jr., J.J.: Relevance feedback in information retrieval. In Salton, G., ed.: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs NJ (1971) 313–323
22. Efron, B.: Bootstrap methods: Another look at the jackknife. Annals of Statistics **7** (1979) 1–26
23. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman and Hall, New York (1993)
24. Wilbur, J.: Non-parametric significance tests of retrieval performance comparisons. Journal of Information Science **20** (1994) 270–284
25. Savoy, J.: Statistical inference in retrieval effectiveness evaluation. Information Processing and Management **33** (1997) 495–512
26. Jijkoun, V., Kamps, J., Mishne, G., Monz, C., de Rijke, M., Schlobach, S., Tsur, O.: The University of Amsterdam at TREC 2003. In: TREC 2003 Working Notes, National Institute for Standards and Technology (2003)

27. Gauch, S., Wang, J.: A corpus analysis approach for automatic query expansion. In: Proceedings of the Sixth International Conference on Information and Knowledge Management, ACM Press (1997) 278–284
28. Gauch, S., Wang, J., Rachakonda, S.M.: A corpus analysis approach for automatic query expansion and its extension to multiple databases. ACM Transactions on Information Systems (TOIS) **17** (1999) 250–269
29. Schütze, H., Pedersen, J.O.: A cooccurrence-based thesaurus and two applications to information retrieval. Information Processing & Management **3** (1997) 307–318
30. Jin, R., Si, L., Hauptman, A.G., Callan, J.: Language model for IR using collection information. In Järvelin, K., Beaulieu, M., Baeza-Yates, R., Myaeng, S.H., eds.: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, New York NY, USA (2002) 419–420
31. Robertson, S.E., Walker, S., Beaulieu, M.: Experimentation as a way of life: Okapi at TREC. Information Processing & Management **36** (2000) 95–108

A Appendix

We analyzed the keyword space using multi-dimensional scaling techniques [15]. The first step is to compute dissimilarities for the controlled vocabulary terms.

A natural candidate for measuring the similarity of the controlled vocabulary terms is the *Jaccard* coefficient. Let $|i|$ denote the number of document having controlled vocabulary term i. For each pair of controlled vocabulary terms i and j, we determine

$$\mathsf{J}(i,j) = \frac{|i \cap j|}{|i \cup j|} = \frac{|i \cap j|}{|i| + |j| - |i \cap j|}.$$

Note that for i we have that $\mathsf{J}(i,i) = 1$ and for disjoint i and j we have $\mathsf{J}(i,j) = 0$. From the Jaccard similarity coefficient, we can make a dissimilarity coefficient by considering $\mathsf{d}_1(i,j) = (1 - \mathsf{J}(i,j))$ or $\mathsf{d}_2(i,j) = \sqrt{(1 - \mathsf{J}(i,j))}$. These dissimilarity coefficients have the following desirable properties, d_1 is metric and d_2 is both metric and euclidean [14].

The Jaccard scores for the collections give values close to 0 for almost all pairs of controlled vocabulary terms. To allow for greater variation, we use the logarithm of the values, thus we determine the distance between two controlled vocabulary terms i and j as

$$\mathsf{Dist}(i,j) = 1 - \frac{\log_{10}(|i \cap j|)}{\log_{10}(|i \cup j|)} = 1 - \frac{\log_{10}(|i \cap j|)}{\log_{10}(|i| + |j| - |i \cap j|)}.$$

This, again, gives a value in the range [0,1], a value 1 for terms not appearing in the same document, a value 0 for terms only occurring in the same documents.

The distance Dist is a metric, i.e, it gives a non-negative number such that

1. $\mathsf{Dist}(i,j) = 0$ if and only if $i = j$,
2. $\mathsf{Dist}(i,j) = \mathsf{Dist}(j,i)$, and
3. $\mathsf{Dist}(i,j) + \mathsf{Dist}(j,k) \geq \mathsf{Dist}(i,k)$.

The third (triangle) inequality will hold due to the fact that all values for distinct i and j are above 0.5.

Based on the above, we can now construct a squared matrix of dissimilarities $\{\mathsf{Dist}(i,j)\}$, of size 6,745 by 6,745 in case of GIRT and of size 10,274 by 10,274 in case of Amaryllis. Our aim is to find a set of points in a lower dimensional space such that each of these points represents one of the controlled vocabulary terms, and that the euclidean distances between points approximate the original dissimilarities as well as possible.

For this, we follow the standard procedure of metric multi-dimensional scaling [15, pp.22–39]. From the dissimilarities, we obtain a matrix **A** of elements $-\frac{1}{2}(\mathsf{Dist}(i,j))^2$. Next, we obtain the double-centered matrix **B**, build from **A** by subtracting row and column mean, and adding matrix mean.

Then spectral decomposition gives

$$\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

with $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ the diagonal matrix of eigenvalues and $\mathbf{V} = (v_1, \ldots, v_n)$ the matrix of corresponding eigenvectors. We assume that the eigenvalues are ordered such that $\lambda_i \geq \lambda_{i+1}$, and that the eigenvectors have unit length.

Following [10], we choose to look at the first 100 eigenvalues $\mathbf{\Lambda}_{100} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{100})$ and associated eigenvectors $\mathbf{V}_{100} = (v_1, \ldots, v_{100})$. The best approximation of **B** on 100 dimensions is matrix $\mathbf{X}_{100}$ such that

$$\mathbf{X}_{100} = \mathbf{V}_{100}\mathbf{\Lambda}_{100}^{1/2}$$

The resulting matrix has dimensions 6,745 by 100 in case of GIRT, and 10,274 by 100 in case of Amaryllis. For each controlled vocabulary term, we now have a vector of length 100.

A Study of the Assessment of Relevance for the INEX'02 Test Collection

Gabriella Kazai, Sherezad Masood, and Mounia Lalmas

Department of Computer Science, Queen Mary University of London
{gabs,shm1,mounia}@dcs.qmul.ac.uk

Abstract. We investigate possible assessment trends and inconsistencies within the collected relevance assessments of the INEX'02 test collection in order to provide a critical analysis of the employed relevance criterion and assessment procedure for the evaluation of content-oriented XML retrieval approaches.

1 Introduction

In information retrieval (IR) research, the evaluation of a retrieval system's performance typically focuses on assessing its retrieval effectiveness. Based on a traditional IR task, such as ad-hoc retrieval, and on a system-centred evaluation viewpoint, effectiveness provides a measure of a system's ability to retrieve as many relevant and as few non-relevant documents to the user's query as possible. Such an evaluation criterion relies on appropriate measures of relevance. In typical IR evaluation experiments, where the predominant approach to evaluate a system's retrieval effectiveness is with the use of test collections, the measure of relevance is provided by human judges. For example, the Text REtrieval Conference (TREC), which is one of the largest evaluation initiatives, employs and trains assessors who then follow guidelines that define relevance and detail the assessment procedure [1].

Traditional IR, however, mainly deals with flat text files. Due to the widespread use of the eXtensible Markup Language (XML), especially the increasing use of XML in scientific data repositories, Digital Libraries and on the Web, brought about an explosion in the development of XML tools, including systems to store and access XML content. The aim of such retrieval systems is to exploit the explicitly represented logical structure of documents, and retrieve document components, instead of whole documents, in response to a user query. Implementing this, more focused, retrieval paradigm means that an XML retrieval system needs not only to find relevant information in the XML documents, but also to determine the appropriate level of granularity to return to the user [14].

A fundamental consequence of the XML retrieval paradigm is that the relevance of a retrieved component is dependent on meeting both content and structural conditions. Evaluating the effectiveness of XML retrieval systems, hence, requires a test collection where the relevance assessments are provided according to a relevance criterion that takes into account the imposed structural aspects. A test collection as such has been built as a result of the first round of the INitiative for the Evaluation of

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 296–310, 2004.

XML Retrieval (INEX)[1]. The initiative was set up at the beginning of 2002 with the aim to establish an infrastructure and provide means, in the form of a large XML test collection and appropriate scoring methods, for the evaluation of content-oriented retrieval of XML documents. In this paper, we make use of the constructed test collection, and in particular the collected relevance assessments. Our aim is to investigate the assessment of relevance for XML documents, explore possible assessment trends both with respect to individual and related components and examine the consistency and exhaustiveness of the assessments. Our study provides a critical analysis of the relevance criterion and assessment procedure employed in INEX’02, which serves as input to future INEX runs.

The paper is organized as follows. In section 2, we discuss the concept of relevance both in general and as defined in INEX’02. In Section 3, we describe the INEX test collection, and the methodology used to obtain the relevance assessments. Our investigation of the collected assessments consists of three parts. First, in Section 4, we examine the distribution of the relevance assessments. Second, in Section 5, we look at the assessments of related elements. Finally, in Section 6, we investigate the consistency and exhaustiveness of the assessments. The paper concludes in Section 7 with guidelines for future runs of INEX.

2 The Concept of Relevance in Information and XML Retrieval

Dictionaries define relevance as “pertinence to the matter at hand”. In terms of IR, it is usually understood as the connection between a retrieved document and the user’s query. With respect to the evaluation of IR systems, relevance plays a fundamental role as “relevance judgments form the bedrock on which the traditional experimental evaluation model is constructed” [2]. Due to its importance in IR theory, the concept of relevance has been the subject of numerous studies over the years. Despite it being a “primitive concept” that people understand intuitively [3], several interpretations of relevance, such as “aboutness” or “utility”, have been explored in the past. Of these, a recent study found topicality as the most important criterion for relevance [4]. However, many researchers agree that relevance is a multidimensional cognitive concept whose meaning is dependent on the users’ perceptions of information [5,6]. Several studies examine these dimensions [7], while others concentrate on defining various manifestations of relevance, e.g. algorithmic, situational, or motivational [8]. Relevance is also considered as a dynamic notion reflecting the finding that a user’s perception of relevance may change over time. A relevance judgement, hence, is described as an assignment of a value of relevance by a judge or assessor at a certain point in time [9]. Furthermore, relevance is described as a multilevel phenomenon, according to which some documents may be more relevant than others [10,11,12].

Despite its many characteristics, relevance is considered a systematic and measurable concept when approached conceptually and operationally from the user’s perspective [6]. It has also been successfully employed in IR evaluations as the standard criteria of evaluation, where research showed that, despite its dynamic nature, which can lead to large differences between relevance judges, the comparative

[1] http//qmir.dcs.qmul.ac.uk/inex/

evaluation of retrieval systems based on test collections provides reliable results when based on large number of topics [13].

In XML retrieval, the relationship between a retrieved item and the user's query is further complicated by the need to consider an additional dimension brought upon by the structural knowledge inherent in the documents and the possible structural conditions specified within the user's query. Given the need to accommodate both content and structural aspects, INEX defined a relevance criterion with the two dimensions of topical relevance and component coverage (both based on the topicality aspect of relevance) [14, pp184]. Topical relevance was defined as a measure of how exhaustively a document component discusses the topic of the user's request, while component coverage was defined as the extent to which a document component focuses on the topic of request (and not on other, irrelevant topics). Topical relevance adopted a four-point ordinal relevance scale based on the one proposed in [11]:

- Irrelevant (0): the document component does not contain any information about the topic of request.
- Marginally relevant (1): the document component refers to the topic of request but only in passing.
- Fairly relevant (2): the document component contains more information than the topic description, but this information is not exhaustive.
- Highly relevant (3): the document component discusses the topic of request exhaustively.

For component coverage, a nominal scale was defined:

- No coverage (N): the topic or an aspect of the topic is not a theme of the document component.
- Too large (L): the topic or an aspect of the topic is only a minor theme of the document component.
- Too small (S): the topic or an aspect of the topic is the main or only theme of the document component, but the component is too small to act as a meaningful unit of information when retrieved by itself.
- Exact coverage (E): the topic or an aspect of the topic is the main or only theme of the document component, and the component acts as a meaningful unit of information when retrieved by itself.

The combination of these two relevance dimensions was used to identify those relevant document components, which were both exhaustive and specific to the topic of request and hence represent the most appropriate unit to return to the user.

3 The INEX'02 Test Collection

The document collection of the INEX'02 test collection consists of the full texts of 12107 articles of the IEEE Computer Society's publications, from 1995 to 2002, totalling 494 megabytes [14, pp1]. On average, an article contains 1532 XML nodes (including attribute nodes), where the average depth of a node is 6.9. The overall structure of a typical article consists of a frontmatter (fm), a body (bdy) and a backmatter (bm). The frontmatter contains the article's metadata, such as title, author, publication information, and abstract. The body is structured into sections (sec) and sub-sections (ss1). These logical units start with a title, and contain a number of paragraphs (para[2]), tables, figures, lists, citations, etc. The backmatter includes bibliography and author information. The "collection" column of Table 1 shows the occurrence frequency of different element types in the collection.

In INEX'02, two types of topics were used: 1) content-only (CO), which are typical IR queries where no constraints are formulated with respect to the structure of the retrieval results, and 2) content-and-structure (CAS), which are XML queries that contain explicit references to the XML structure, i.e. specifications of target elements (e.g. what should be returned to the user) and containment conditions (e.g. element types that should be about a given concept).

During the retrieval sessions participating groups produced a ranked list of XML elements in answer to a topic. The top 100 result elements from all 60 sets of ranked lists (one per topic) formed the results of one retrieval run. A total of 51 runs from 25 groups were submitted. For each of the 60 topics, the results from the submissions were merged to form the pool for assessment (see Table 1, "Result pool" columns).

The result pools were then assigned for assessment either to the original topic authors or, when this was not possible, on a voluntary basis, to groups with expertise in the topic's subject area. The assessments were done along the two dimensions of topical relevance and component coverage. Assessors were asked to judge each and every relevant document component by following a two-step process [14, pp184]. During the first step, assessors were required to skim-read the whole article that contained a result element and identify any relevant information within. In the second step, assessors had to judge the relevance of the found relevant components and of their ascendant and descendant elements. Assessors were allowed to stop assessing ascendant elements once a container component was judged as too large. Similarly, descendant elements only needed to be judged until an irrelevant component or a component with too small coverage was reached. To lessen the workload, the system implicitly regarded any non-assessed elements as irrelevant. For CAS topics with target elements, the procedure was modified stating that any elements other than the target elements had to be considered irrelevant.

Assessments were recorded using an on-line assessment system, which allowed judges to view the result pool of a given topic (listing result elements in alphabetical order), browse the document collection and view articles and result elements both in XML (i.e. showing the tags) and document view (i.e. formatted for ease of reading). Other features included keyword highlighting and consistency checking.

Assessments were collected for 55 (30 CAS and 25 CO) of the 60 topics, for a total of 48849 files containing 71086 elements, of which 22719 are at article level. The last three columns of Table 1 show a breakdown of the collected assessments by element type for both topic types. Note that these statistics only include explicit assessments, i.e. implicitly irrelevant elements are not considered. In addition, the assessments of 8246 article files for CAS and 10717 for CO are also excluded. The reason for this is that these files were assessed using a "quick assess" option of the on-line assessment system, which allowed judges to skip the explicit assessment of result elements in an article and only mark the article file as irrelevant.

[2] Paragraphs are elements of the "para" entity as defined in the document collection's DTD: <!ENTITY % para "ilrj|ip1|ip2|ip3|ip4|ip5|item-none|p|p1|p2|p3">.

Table 1. Number of element types in collection, result pool and assessments

Component	Collection	Result pool			Assessed		
		CAS	CO	Total	CAS	CO	Total
Article files	12107	23375	30275	53650	24237	24612	48849
All elements	**8239997**	**47419**	**60066**	**107485**	**34130**	**36956**	**71086**
article	12107	12418	22630	35048	10379	12340	22719
bdy	12107	1215	4133	5348	526	2231	2757
sec	69735	4182	7329	11511	2004	3999	6003
ss1	61492	726	1313	2039	492	1425	1917
para	983737	5349	10133	15482	3645	7505	11150
d.o.[3] para	2835975	1723	1828	3551	2336	1227	3563
fm	12107	4439	2639	7078	1325	862	2187
d.o. fm	383575	7188	1472	8660	4270	1041	5311
bm	10065	268	463	731	361	501	862
d.o. bm	2483446	8175	6066	14241	6460	3632	10092

From Table 1, we can obtain that for CAS 0.58% and for CO 0.73% of all XML elements in the collection were included in the result pools. The difference between the sizes of the CAS and CO result pools is due to the fact that more runs were submitted for CO topics, where the runs also contained on average more results. On average, 2.0 elements were retrieved from an article file, 68% of which were related by way of ascendant, descendant or sibling relations. Looking at the distribution of the elements in the result pools according to their element types, we can see that article (32.61%), section (10.71%), paragraph (12.07%) and bibliography sub-components (13.25%) were the most frequently returned elements. Another trend that can be observed is that the result pools of CAS topics consisted of approximately equal numbers of "small" (i.e. para, sub-components of para, fm and bm) and "large" (i.e. article, sec, ss1, fm and bm) components, while for CO topics this ratio is around 35% and 65%, respectively. From the result pools, 72% of the results were explicitly assessed for CAS and 61.5% for CO. Note that this only indicates that assessors more often used the "quick assess" option with CO topics (all results were assessed and even some additional article files that were not included in the result pool).

4 Investigation of the INEX'02 Relevance Assessments

In this section we investigate the distribution of the collected assessments at different levels of granularity in order to derive conclusions of possible assessment trends.

4.1 Distribution of Topical Relevance

We first look at the distribution of assessments for the topical relevance dimension (Table 2). In general, for both topic types a large proportion of the results were judged irrelevant (71% for CAS and 48% for CO) and only a small portion were perceived as highly relevant (9% and 8%). This is not surprising and it correlates with findings of previous studies [15]. What is interesting, however, is the relatively high percentage

[3] d.o. = descendant elements of ...

of irrelevant CAS assessments compared with CO. The reason for this lies in the definition of the CAS relevance criterion, which explicitly states that any components other than target elements must be treated as irrelevant. Since a number of CAS submissions were produced by systems that did not support XML style queries, a high percentage of the CAS result elements were hence irrelevant, non-target elements.

With respect to the highly relevant elements, although their numbers are the same for CAS and CO, they represent a very different distribution, making up for CAS 32% and for CO 16% of the total number of their respective relevant elements. These ratios are also magnitudes higher in INEX than in flat text test collections [15]. This is mainly due to the structure of the collection and the definition of the relevance criterion. As mentioned earlier, assessors in INEX had to assess both ascendant and descendant elements of a relevant component. Furthermore, due to the definition of topical relevance, we know that the relevance degree of an ascendant node is always equal to or greater than the relevance degree of its sub-nodes. This means that any elements assessed as highly relevant will have highly relevant ancestor nodes. This propagation effect of topical relevance, combined with its cumulative nature, provides an answer to the increased level of highly relevant elements for CO topics.

For CAS topics, however, no such relation between the relevance degrees of related components exists (as only target elements are relevant). Furthermore, looking at the ratio of marginally and fairly relevant CAS assessments, we can see that, while the proportion of marginally relevant components is the same as for CO topics (41%), the relatively high ratio of highly relevant elements is complemented with a low percentage of fairly relevant elements (27%). A plausible reason, given the question-answering nature of most CAS topics and that 47% of CAS result elements were small components, is that the degree of fairly relevant was less discernable (or less meaningful) in this context. By definition, an element should be assessed fairly relevant if it contains more information than the topic description, but this information is not exhaustive. It is not clear, however, when the content of, for example, an author element would match such a criterion. It is hence more likely that in these cases assessors assigned either marginally or highly relevant degrees. To further investigate this issue, Table 3 shows the averaged relevance distributions for the different categories of CAS topics: those that do not specify target elements; those where the requested result elements are of factual type (e.g. author, title, bibliographic entry); or content-oriented (e.g. article, sec, para). As it can be seen, the distribution of the assessments for CAS topics with no target element closely follows the distribution of CO assessments, while the assessments of CAS topics with target elements demonstrates a very different behaviour. For factual result elements, we find that a dominant percentage were assessed as marginally relevant (52%), while for content-oriented results the ratio of highly relevant assessments is the dominant (51.1%). A reason for the former finding is the high percentage of small result components (e.g. title), whose "exhaustiveness"-level was assessed to satisfy only the minimum criteria of topical relevance. Although no definitive reasoning can be given for the latter finding, we suspect that judges were partly influenced by a match regarding the target element specification (i.e. were partial to judge an element highly relevant if it matched the target element). This is further supported when looking at the breakdown of the assessments for the different element types in Table 2. Looking at the CAS columns, we can see that the majority of highly relevant components (57.7%) consist of articles and sub-components of fm and bm, which also represent over 60% of the target element types of CAS topics.

Table 2. Distribution of topical relevance assessments for CAS and CO topics

		CAS					CO				
rel =		3	2	1	0	Total	3	2	1	0	Total
Total		3102 **9%**	2703 **8%**	3980 **12%**	24345 **71%**	34130 100%	3026 **8%**	8191 **22%**	7844 **21%**	17895 **48%**	36956 100%
Rel. only		**32%**	**27%**	**41%**	-	9785	**16%**	**43%**	**41%**	-	19061
article	(%)	**11.6**	8.0	9.3	**38.7**	30.4	**23.2**	10.8	13.8	**54.0**	33.4
bdy	(%)	3.5	3.7	2.1	1.0	1.5	14.6	6.9	7.5	3.5	6.0
sec	(%)	9.1	12.8	8.2	4.3	5.9	**19.8**	**15.7**	12.6	6.3	10.8
ss1	(%)	3.2	4.4	1.7	0.9	1.4	6.8	5.9	5.4	1.8	3.9
para	(%)	8.7	**22.4**	**14.5**	9.0	10.7	**18.7**	**30.7**	**28.0**	12.4	20.3
d.o. para	(%)	2.0	5.3	9.4	7.2	6.8	0.4	3.3	3.6	3.7	3.3
fm	(%)	1.1	2.2	2.0	4.7	3.9	1.6	1.8	1.5	3.1	2.3
d.o. fm	(%)	**28.2**	11.4	7.5	11.5	12.5	3.1	3.5	2.9	2.4	2.8
bm	(%)	0.9	1.2	0.8	1.1	1.1	1.5	1.8	2.4	0.7	1.4
d.o. bm	(%)	**17.9**	**23.5**	**36.3**	**15.7**	18.9	5.1	12.8	**15.3**	6.9	9.8

Table 3. Distribution of topical relevance assessments for CAS with different types of target elements

	rel=	3	2	1
Factual target element	(%)	27.9	20.1	**52.0**
Content target element	(%)	**51.1**	20.3	28.6
No target element	(%)	22.9	40.0	37.1

In contrast, for CO topics, the majority of highly relevant elements are articles, sections and paragraphs (61.7%), and fairly and marginally relevant elements are mostly paragraphs, sub-components of bm, and sections (59.2% and 55.9%). At first glance this would suggest a clear preference for larger components for CO topics. This is, however, not the case, but these findings simply show the propagation effect of topical relevance and confirm its cumulative nature.

4.2 Distribution of Component Coverage

Table 4 summarises the distribution of assessments with respect to the component coverage dimension. Looking at the totals, a noticeable difference is the relatively high ratio of exact coverage (16%) and the relatively low ratio of too large (7%) and too small (6%) assessments for CAS topics, compared with CO (10%, 22% and 20%, respectively). Looking at the distribution of relevant elements only, we can observe a very high ratio of exact coverage for CAS (57%) compared with CO (19%). A reason for this is that possibly relevant, but non-target elements, which may otherwise had been assessed as too large or too small, were judged irrelevant due to the strict CAS relevance criterion. However, we suspect that another reason for the domination of exact coverage assessments is that assessors incorrectly assigned exact coverage to elements that matched the target element of the topic (instead of assessing components according to the extent to which they focus on the topic of the request). This concern was also verbally confirmed by some of the assessors at the INEX'02 workshop. The root of the problem is that the "too small" and "too large" coverage categories were incorrectly interpreted as the relation between the actual size of the result component and the size of the target element (instead of the relation between the relevant and irrelevant contents of the component). Further evidence of this can be seen in Table 5, which shows the averaged distribution of coverage assessments for CAS topics with and without target elements. Although for factual result elements it is reasonable to expect higher levels of exact coverage assessments, the distribution for content-oriented results is expected to closer follow the distribution of CAS topics with no target element. This is because while it is plausible that an author element, for example, contains only relevant information regarding a query asking for names of experts in a given field, it is less likely that target elements, such as sections or articles, contain no irrelevant information to the topic of the request.

Table 4. Distribution of component coverage assessments for CAS and CO topics

	CAS					CO				
cov=	E	L	S	N	Total	E	L	S	N	Total
Total	5530 **16%**	2214 **7%**	2041 **6%**	24345 **71%**	34130 100%	3611 **10%**	8219 **22%**	7231 **20%**	17895 **48%**	36956 100%
Rel. only	**57%**	**22%**	**21%**	-	9785	**19%**	**43%**	**38%**	-	19061
article (%)	5.9	**24.7**	3.7	**38.7**	30.4	**16.3**	**22.5**	3.2	**54.0**	33.4
bdy (%)	0.5	11.4	0.6	1.0	1.5	4.6	**16.5**	1.0	3.5	6.0
sec (%)	6.6	**22.3**	4.7	4.3	5.9	**16.7**	**18.3**	10.6	6.3	10.8
ss1 (%)	1.7	6.2	2.4	0.9	1.4	9.0	6.3	3.7	1.8	3.9
para (%)	11.3	7.8	**32.2**	9.0	10.7	**35.5**	9.8	**44.2**	12.4	20.3
d.o. para (%)	3.1	0.1	**20.0**	7.2	6.8	1.6	0.8	6.2	3.7	3.3
fm (%)	0.5	3.5	3.3	4.7	3.9	0.8	2.4	1.3	3.1	2.3
d.o. fm (%)	**22.4**	5.2	6.3	11.5	12.5	5.0	2.1	3.4	2.4	2.8
bm (%)	0	3.5	0.7	1.1	1.1	0.2	4.1	0.5	0.7	1.4
d.o. bm (%)	**40.1**	10.2	9.5	**15.7**	18.9	9.3	10.1	**17.2**	6.9	9.8

Table 5. Distribution of component coverage assessments for CAS with different types of target elements

rel=	E	L	S
Factual target element (%)	**76.4**	9.7	13.9
Content target element (%)	**56.0**	21.8	22.2
No target element (%)	27.1	39.7	33.2

The distribution of the coverage assessments according to element types (Table 4) shows that for CAS topics the most common elements with exact coverage (73.8%) were paragraphs and sub-components of bm and fm, while articles and sections were mostly judged as too large (47%). For CO topics, we can observe a clear preference towards paragraphs being judged with exact coverage (35.5%), while a large proportion of articles and sections were assessed as too large (40.8%). Although this finding may not be so surprising, it clearly demonstrates that judges preferred more specific elements for both topic types. It also correlates with expectations that purely relevant information is likely to be contained in smaller units and that larger nodes, such as articles, are more likely to include other, irrelevant information. Combined with the relevance distribution data ("Rel. only" row), these results suggest that for CO topics, assessors were able to interpret the coverage criterion correctly and assess components accordingly. Looking at the too small assessments, we find that the majority of element types for CAS are paragraphs and sub-components of paragraphs (52.2%). An interesting observation here is that although a high percentage of paragraphs were found to be too small, their container components, such as sec, bdy and article elements were generally found too large, leaving no elements in between with exact coverage. A reason for this is that container elements, which may otherwise had exact coverage of the topic, were implicitly regarded as irrelevant if they did not match the target element. For CO topics, the distribution of too small assessments is more clear-cut. Here, small elements, e.g. paragraphs and sub-components of bm make up 61.4% of the too small assessments. It should be noted that here the large proportion of too small assessments is complemented with high proportions of exact coverage article, sec and ss1 elements (42%).

4.3 Distribution of the Combined Relevance Assessments

In this section we investigate the correlation between the relevance dimensions. Although the two dimensions are not completely independent (i.e. combinations like 0E, 0L, 0S, 1N, 2N and 3N are not permitted and assessments of 3S are not reasonable), strong correlations in the following analysis would indicate the presence of common factors that influenced the assessment of both dimensions.

Table 6 shows the distribution and correlation of the combined assessments for CAS and CO topics. For each possible combination of topical relevance (columns), *rel,* and component coverage (rows), *cov,* the number of assessed components is shown in the "Total" rows (the percentage values reflect the ratio of these elements to all relevant CAS and CO components, respectively). The "cov|rel correlation" rows indicate the correlation of coverage categories with given relevance degrees, where the percentage values represent the likelihood of an element being assessed with coverage level *cov,* given that it has a relevance degree *rel.* Similarly, the "rel|cov correlation" rows indicate the correlation of topical relevance degrees with given coverage levels, where the percentage values represent the likelihood of an element being assessed with relevance degree *rel,* given that it has a coverage level *cov.* As it can be seen, for CAS topics there is a very high likelihood of highly relevant elements being assessed with exact coverage (80.3%). This implies that the assessment of both dimensions was influenced by a common aspect. As we saw in the previous sections, this influencing factor was whether the result element matched the target element of the query. A similar, but less dominating, tendency can be observed for fairly and marginally relevant components (43.9% and 46.5%). Looking at the correlation of coverage given topical relevance, it appears more likely that an element with exact coverage would be judged highly relevant (45%) than fairly (21.5%) or marginally relevant (33.5%). For too large coverage the dominant relevance degree is marginally relevant (43.1%), while for too small coverage an element is almost equally likely to be assessed as fairly or marginally relevant.

For CO topics, no significant correlations are visible. Highly relevant components are just as likely to be assessed to have exact coverage (46.1%) as being judged too large (53.9%). Fairly and marginally relevant components are slightly more likely to be assessed as too small (48.2% and 41.8%) or too large (36.2% and 46.2%) than with exact coverage (15.6% and 12%). These tendencies, however, are mainly due to reasonable correlations between the relevance dimensions and component size. For example, small elements are less likely to be exhaustive or act as meaningful units of information, while large components are more likely to be exhaustive, but also likely to cover multiple topics. The same can be said when looking at the correlation of component coverage given topical relevance.

Table 6. Distribution and correlation of combined assessments for CAS and CO topics

rel:		3		2		1	
cov:		CAS	CO	CAS	CO	CAS	CO
E	Total	2491(25%)	1396(7%)	1187(12%)	1274(7%)	1852(19%)	941(5%)
	cov|rel %	**80.3**	46.1	**43.9**	15.6	**46.5**	12.0
	rel|cov %	**45.0**	38.6	21.5	35.3	33.5	26.1
L	Total	611 (6%)	1630 (9%)	648(7%)	2963(16%)	955(10%)	3626(19%)
	cov|rel %	19.7	53.9	24.0	36.2	24.0	**46.2**
	rel|cov %	27.6	19.8	29.3	36.1	**43.1**	44.1
S	Total	-	-	868(9%)	3954(21%)	1173(12%)	3277(17%)
	cov|rel %	-	-	32.1	**48.2**	29.5	41.8
	rel|cov %	-	-	42.5	54.7	57.5	45.3

5 Investigating the Assessment of Related Components

Since document components returned by an XML retrieval system may be related we cannot regard their relevance as independent from one another. In this section we examine the relevance values assigned to related components, such as parent, child, sibling, ancestor and descendant nodes. Our aim is to discover possible correlations and identify factors, if any, that influence the assessment of related components.

5.1 Topical Relevance

Table 7 lists the occurrence frequency of assessment pairs for each possible combination of element and related element assessments of topical relevance (e.g. 0,0; 0,1; 0,2; 0,3; etc.) for each relation type (element,parent; element,child; etc) and for both CO and CAS topic types.

We first look at the results for CO topics. Due to the definition of topical relevance, we expect to find certain regularities in the assessed relationships, such as the equivalence or increase of the assigned relevance degree for parent and ascending nodes. This is clearly visible, apart from some noise due to assessment mistakes (see section 6), at all levels of topical relevance. In addition, we can observe a tendency to assign equal, instead of higher, degrees of relevance (approx. 60% to 40%) to parent/ancestor nodes. This tendency is particularly strong for the element-parent relations and for the assessment pairs 1,1 (73.09%) and 2,2 (80.43%). An explanation for this is the propagation effect of topical relevance. Other factors include situations where the available relevance degrees are insufficient in reflecting a change in the amount of relevant information contained by a parent/ancestor node. On the other hand, when the relevance level of parent or ascendant nodes is increased, the increase tends to be as minimal as possible, e.g. it is more likely that the parent of a marginally relevant node is assessed as fairly relevant (1,2: 21.53%) rather than highly relevant (1,3: 5.34%). More or less the same observations apply, as expected, to the element-child and element-descendant relations with only slight differences in the actual percentage values. Looking at the element-sibling relations, we are not able to find such clear patterns in the assessments. The trend of assigning related components the same relevance degree seems to hold, but is less prominent (e.g. 3,3: is only 48.31%). Also, although there is still evidence that the relevance level of sibling nodes is more likely to differ from the relevance of a given node by only one degree, this trend does not hold for highly relevant elements (i.e. 3,0: 12.92%; 3,1: 9.18%).

Table 7. Topical relevance assessments of related components for CAS and CO topics

Relation=	parent		child		sibling		ascendant		descendant	
(%)	CAS	CO	CAS	CO	CAS	CO	CAS	CO	CAS	CO
0,0	68.52	46.18	94.48	99.87	93.87	68.07	67.58	38.90	94.90	99.83
0,1	7.34	26.23	1.56	0.13	2.24	19.49	4.40	24.47	2.34	0.17
0,2	11.02	18.21	0.73	0	2.24	9.35	11.23	19.47	1.52	0
0,3	13.12	9.38	3.24	0	1.65	3.10	16.80	17.16	1.25	0
Subtotal	65.30	23.51	47.36	10.87	69.16	24.44	71.28	24.69	50.76	9.62
1,0	5.63	0.05	40.42	21.05	10.13	16.76	9.85	0.05	40.05	24.31
1,1	53.52	**73.09**	59.08	78.70	73.26	62.47	37.84	58.69	58.26	75.32
1,2	31.24	**21.53**	0.42	0.23	14.62	18.88	32.11	26.69	1.36	0.35
1,3	9.60	**5.34**	0.07	0.02	1.98	1.89	20.20	15.01	0.33	0.02
Subtotal	13.09	31.53	11.85	29.29	15.31	28.42	12.05	31.90	7.82	24.86
2,0	2.69	0	33.69	11.07	13.37	5.54	7.21	0	**42.58**	13.25
2,1	0.39	0.20	19.14	17.56	19.29	13.00	0.99	0.25	20.57	23.08
2,2	78.62	**80.43**	47.02	71.30	63.48	77.27	64.24	66.10	36.56	63.57
2,3	18.30	19.37	0.16	0.07	3.86	4.20	27.57	33.65	0.29	0.11
Subtotal	12.78	34.28	21.37	38.67	11.60	41.28	10.70	34.89	18.80	36.28
3,0	17.36	0	**44.11**	7.95	29.01	**12.92**	10.62	0	**52.94**	14.49
3,1	0.09	0.04	6.47	10.42	7.72	**9.18**	0.43	0.06	10.76	16.37
3,2	0.38	0.27	12.04	31.36	11.39	29.59	0.91	0.45	13.04	29.00
3,3	82.16	99.69	37.38	50.27	51.89	**48.31**	88.04	99.49	23.26	40.14
Subtotal	8.83	10.68	19.42	21.17	3.93	5.86	5.97	8.52	22.62	29.24

Looking at the results for CAS topics, none of the above trends can be observed clearly. For example, although the tendency to assign the same relevance degree to related components is still present, there are several cases where this pattern does not apply (e.g. element-child of 3,0: 44.11%; element-descendant of 2,0: 42.58% and 3,0: 52.94%,). The pattern that the relevance degree of related nodes usually only differs by one degree does not apply at all, but the assessment pairs appear more random, although there is a noticeable increase in the 1,0; 2,0 and 3,0 assessment pairs for all relation types. This is again related to the strict relevance criterion for CAS topics.

5.2 Component Coverage

Table 8 shows the occurrence frequency of assessment pairs for each possible combination of element and related element assessments of component coverage (e.g. N,N; N,L; N,S; etc.) for each relation type and for both topic types.

Several general trends can be observed for CO topics. The most obvious perhaps is that 90.62% of parent components of too large elements are also assessed as too large. Although this is largely expected, it cannot be applied as a general rule since the ratio of relevant information contained in a parent node is also dependent on the sibling elements' contents. This is reflected in the finding that 6.39% of parent nodes of too large elements have been judged to have exact coverage. The fact that the coverage of ascendant nodes of too large components cannot be inferred with 100% accuracy highlights a false assumption in the assessment procedure. According to the instructions, assessors were allowed to stop the assessment of ascendant nodes once a too large component was reached, assuming that all ascendants would also then be too large. The same applies regarding the stopping rule of assessing descendant elements. Although most child nodes of too small elements are also assessed as too small (84.48%), this is not a certainty, e.g. child nodes may also be irrelevant (8.31%) or too large (6.77%). Other assessment patterns in Table 8 include the high correlation of sibling nodes assessed with the same coverage degree (N,N 68.07%, L,L 55.38%, S,S 86.57% and E,E 49.24%).

Similarly to CO, for CAS topics the parent nodes of too large elements are also assessed as too large (91.47%). However, these assessment pairs comprise only 9.78% of all CAS assessment pairs (i.e. CAS topics with no target element). Another pattern is the relative increase in the number of L,N; S,N and E,N element-child and element-descendant relations due to the already mentioned strict relevance criterion.

Table 8. Component coverage assessments of related components for CAS and CO topics

Relation=	parent		child		sibling		ancestor		descendant	
(%)	CAS	CO	CAS	CO	CAS	CO	CAS	CO	CAS	CO
N,N	68.52	46.18	94.48	99.87	93.87	**68.07**	67.58	38.90	94.90	99.83
N,L	15.75	40.71	1.17	0.13	1.00	8.51	21.08	50.32	1.28	0.13
N,S	3.81	4.63	0.46	0	2.10	15.58	1.59	2.30	1.73	0.03
N,E	11.92	8.48	3.89	0	3.03	7.85	9.75	8.48	2.09	0
Subtotal	65.66	23.51	47.36	10.87	69.16	24.44	71.28	24.69	50.76	9.62
L,N	0	0	**37.59**	17.85	35.10	18.41	10.23	0.06	**47.43**	19.77
L,L	**91.47**	**90.62**	32.54	50.17	43.10	**55.38**	81.39	86.96	16.34	31.05
L,S	3.19	2.99	11.37	16.01	4.92	12.84	3.78	3.66	19.52	32.01
L,E	5.34	6.39	18.50	15.97	16.87	13.37	4.60	9.32	16.71	17.16
Subtotal	**9.78**	29.70	27.35	53.63	1.98	11.30	6.36	22.44	31.68	62.84
S,N	1.65	0	**48.61**	**8.31**	16.59	7.09	6.57	0.01	**43.74**	6.85
S,L	23.51	26.11	6.06	**6.77**	1.11	2.70	46.30	51.76	9.26	9.92
S,S	17.36	33.65	44.84	**84.48**	75.48	**86.57**	8.70	17.66	44.74	82.81
S,E	57.48	40.24	0.49	0.44	6.82	3.63	38.43	30.57	2.27	0.43
Subtotal	13.30	32.89	5.12	13.10	8.75	53.67	13.36	38.86	2.60	8.29
E,N	16.47	0	**38.60**	8.90	10.42	18.10	11.81	0	**46.45**	10.88
E,L	45.21	61.61	2.58	8.47	1.66	14.26	58.80	76.98	1.95	10.86
E,S	0.22	0.41	37.68	59.06	2.97	18.39	0.65	0.25	34.31	61.70
E,E	38.10	37.98	21.15	23.56	84.95	**49.24**	28.73	22.77	17.29	16.57
Subtotal	11.26	13.90	20.17	22.40	20.12	10.59	9.00	14.01	14.96	19.26

6 Exhaustiveness and Consistency of the Assessments

The assessment procedure stated that for topics without target elements all ascendants and descendants of each relevant component should be assessed until a too large ascendant or a too small or irrelevant descendant element is reached. The assessment of too large nodes was then propagated to ascendants, while all other non-assessed elements were implicitly regarded as irrelevant. Due to these implicit rules, we have only limited means by which to estimate the exhaustiveness of the assessments. For topics with target elements, we have no way of checking if all necessary elements have been assessed due to the strict relevance constraint and the implicitly irrelevant assumption. For the remaining topics, we calculated:

- The number of relevant elements that only have irrelevant descendants: we found only 15 elements (8 2S and 7 1E);
- The number of relevant elements with exact or too small coverage that do not have an ancestor with too large coverage: we obtained 29 highly relevant, 62 fairly relevant and 86 marginally relevant elements;
- The number of elements whose siblings have not been assessed, but whose parent node has a higher level of relevance: there are 229 fairly relevant and 501 marginally relevant elements.

These figures appear very comforting, however with no exact way to confirm other possibly missing assessments we have to consider these only as very rough estimates of assessment exhaustiveness. In addition, circumstantial evidence, in the form of lack of overlap within the submission results (see Table 1), suggests that further relevant components may not have been retrieved and hence assessed.

Table 9. Inconsistent assessment pairs for CO topics

	0,1	0,2	0,3	1,0	1,2	1,3	2,0	2,1	2,3	3,0	3,1	3,2
parent	-	-	-	3	-	-	-	14	-	-	1	6
child	3	-	-	-	14	1	-	-	6	-	-	-
ancestor	-	-	-	10	-	-	-	54	-	-	3	24
descendant	10	-	-	-	54	3	-	-	24	-	-	-

Next, we examine the consistency of the collected assessments. Since the on-line assessment tool already included consistency checking for the individual assessments (such as checking for invalid assessments, e.g. 0E, 0L, 0S, 1N, 2N, 3N and 3S), here we concentrate on the consistency of related assessments. We again deal only with topics that do not specify target elements. In this context, we consider an assessment pair inconsistent when an ancestor of a relevant node is assessed less relevant or irrelevant; or vice-versa when a descendant is assessed as more relevant. Table 9 shows the total number of inconsistencies found. Note that here we only show data based on 4 topics that were assessed before a version of the on-line assessment tool included rules to test the consistency of related assessments. Although our observations here are not statistically significant, we can still conclude that the number of inconsistencies (91) compared with the total number of assessed components (7340, including 3327 relevant) is largely insignificant. As it can be seen, most of the inconsistencies occur as a result of ancestors of fairly relevant node's being assessed only as marginally relevant (59%), where of the 54 cases 14 are direct element-parent relations. In general, 97% of the inconsistencies are assessments where the ancestor node of the element is judged one degree less relevant than the node itself. A possible reason for the inconsistencies is that in the XML view of the on-line assessment tool large articles required a lot of scrolling up and down to assess parent and ancestor nodes, where assessors could have easily missed some relevance values assigned to sub-components.

7 Conclusions

In this paper we provided an analysis of the relevance assessments of the INEX'02 test collection; examined their distributions over the two dimensions at different levels of granularity; explored the assessments of related components; and investigated assessment consistency and exhaustiveness.

Our findings, in general, showed no surprises, but confirmed expected effects of the relevance criterion (such as the strict criterion for CAS topics, the cumulative nature of topical relevance for CO topics and its propagation effect to ancestor nodes). We found that the combination of the two dimensions worked well for CO topics allowing assessors to identify elements both exhaustive and specific to the topic of request. For topical relevance, the number of relevance degrees was found suitable, although we also found indications that both less and more degrees would have been useful in different situations. An issue regarding the degree of fairly relevant was highlighted by the finding that it appeared less measurable for small factual results. This is because its definition is based on the notion of exhaustivity, which is more suited for content-oriented, and in particular multifaceted, topics. In general, however, the dimension of topical relevance was found appropriate for XML evaluation, even though it does not consider aspects of usefulness or novelty (which are important factors given that retrieved elements may be nested components). Our main criticism regards the coverage dimension and the assessment of CAS topics with target elements. We found evidence to show that a match on a target element type influenced the assessment of both relevance dimensions, and especially the dimension of component coverage. Furthermore, we reported on the concern that the categories of too large and too small were also misinterpreted when target elements were assessed. A general issue regarding the too small coverage was that it combined criteria regarding both topicality and unit size, which actually encouraged its misinterpretation. This issue has since been addressed in INEX'03[4], where the component coverage dimension was redefined to avoid direct association with component size. A solution regarding the assessment of coverage of target elements, where assessors are instructed to ignore the structural constraints within the query, is also being tested in INEX'03.

Regarding the consistency of the assessments we found no reason to warrant concern, even without the consistency checking tools of the on-line assessment system. On the other hand, there is cause for concern regarding the exhaustiveness of the assessments especially as the overlap between retrieval submissions is low. This is addressed in INEX'03 by means of rigorous checks being implemented in the assessment tool, where assessors are required to assess all relevant ascendant and descendant components.

Regarding the future of XML evaluation, it is clear that as XML is becoming more popular, the need for test collections also increases. The study of the construction of such test collections, however, is still in its infancy. The first year of the INEX initiative developed and tested a new criterion for relevance based on two dimensions. Our findings in this paper highlighted some issues and concerns regarding the definition of both this criterion and the assessment procedure. Possible solutions to these problems are currently put to the test in INEX'03.

[4] http://inex.is.informatik.uni-duisburg.de:2003/

References

[1] Voorhees, E.M. and Harman, D.K., eds. (2002): The tenth Text Retrieval Conference (TREC-2001), Gaithersburg, MD, USA, 2002. NIST.

[2] Harter, S.P. (1996): Variations in Relevance Assessments and the Measurement of Retrieval Effectiveness. Journal of the American Society for Information Science, 47(1):37-47.

[3] Saracevic, T.: www.scils.rutgers.edu/~tefko/Courses/610/Lectures/.

[4] Vakkari, P. and Hakala, N. (2000): Changes in Relevance Criteria and Problem Stages in Task Performance. Journal of Documentation. 56(5): 540-562.

[5] Schamber, L. (1994): Relevance and Information Behaviour. Annual Review of Information Science and Technology (ARIST), 29:3-48.

[6] Borlund, P. (2003): The Concept of Relevance in IR. Journal of the American Society for Information Science, 54(10):913-925.

[7] Cosijn, E. and Ingwersen, P. (2000): Dimensions of Relevance. Information Processing and Management, 36:533-550.

[8] Saracevic, T. (1996): Relevance Reconsidered. Proceedings of the 2nd International Conference on Conceptions of Library and Information Science (COLIS), Copenhagen, pp. 201-218.

[9] Mizzaro, S. (1997): Relevance: the whole history. Journal of the American Society for Information Science, 48(9):810-832.

[10] Tang, R., Shaw, W.M., and Vevea, J.L. (1999): Towards the identification of the optimal numbers of relevance categories. Journal of the American Society for Information Science, 50(3):254-264.

[11] Kekäläinen, J. and Järvelin, K. (2002): Using graded relevance assessments in IR evaluation. Journal of the American Society for Information Science, 53(13):1120-1129.

[12] Voorhees, E. (2001): Evaluation by highly relevant documents. Proceedings of the 24th ACM-SIGIR conference on research and development in information retrieval, New York, pp. 74-82.

[13] Vorhees, E.M. (1998): Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness. Proceedings of the 21st ACM-SIGIR conference on research and development in information retrieval, Melbourne pp. 315-323.

[14] Fuhr, N., Lalmas, M., Kazai, G., Gövert, N. eds (2003): Proceedings of the first workshop of the INitiative for the Evaluation of XML Retrieval (INEX), Dagstuhl. ERCIM workshop proceedings.

[15] Järvelin, K., Kekäläinen, J. (2000): IR evaluation methods for retrieving highly relevant documents. Proceedings of the 23rd ACM-SIGIR conference on research and development in information retrieval, Athens, pp. 41-48.

A Simulated Study of Implicit Feedback Models

Ryen W. White[1], Joemon M. Jose[1], C.J. van Rijsbergen[1], and Ian Ruthven[2]

[1] Department of Computing Science,
University of Glasgow. Glasgow, Scotland. G12 8RZ.
{ryen,jj,keith}@dcs.gla.ac.uk
[2] Department of Computer and Information Sciences,
University of Strathclyde. Glasgow, Scotland. G1 1XH.
ir@cis.strath.ac.uk

Abstract. In this paper we report on a study of implicit feedback models for unobtrusively tracking the information needs of searchers. Such models use relevance information gathered from searcher interaction and can be a potential substitute for explicit relevance feedback. We introduce a variety of implicit feedback models designed to enhance an Information Retrieval (IR) system's representation of searchers' information needs. To benchmark their performance we use a simulation-centric evaluation methodology that measures how well each model learns relevance and improves search effectiveness. The results show that a heuristic-based binary voting model and one based on Jeffrey's rule of conditioning [5] outperform the other models under investigation.

1 Introduction

Relevance feedback (RF) [11] is the main post-query method for automatically improving a system's representation of a searcher's information need. The technique relies on explicit relevance assessments (i.e. indications of which documents contain relevant information), and creates a revised query attuned to those documents marked. The need to explicitly mark relevant documents means searchers may be unwilling to directly provide relevance information.

Implicit RF, in which an IR system unobtrusively monitors search behaviour, removes the need for the searcher to explicitly indicate which documents are relevant. The technique uses implicit relevance indications, gathered from searcher interaction, to modify the initial query. Whilst not being as accurate as explicit feedback, in previous work [14] we have shown that implicit feedback can be an effective substitute for explicit feedback in interactive information seeking environments. In this paper we evaluate the search effectiveness of a variety of implicit models using a simulation-based methodology. This strategy, similar to [6,9], is not affected by inter-searcher inconsistencies, is less time consuming and costly, and allows environmental and situational variables to be more strictly controlled. It allows us to compare and fine- tune the various models before they are employed in a real system. We use simulations since no precedent has yet been set on how to best evaluate implicit feedback models.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 311–326, 2004.

We investigate a variety of different methods of relevance feedback weighting based on implicit evidence. The implicit feedback models presented use different methods of handling this implicit evidence and updating their understanding of searcher needs in light of it. The study compares the models' ability to learn relevance and create more effective search queries.

The remainder this paper is structured as follows. In Section 2 we describe the document representations and relevance paths used to create evidence for the models described in Section 3. In Section 4 we describe the simulations used to test our approach, the results in Section 5, and conclude in Section 6.

2 Document Representations and Relevance Paths

The implicit models we evaluate in this paper gather relevance information from searchers' exploration of the *information space;* the information content of the top-ranked retrieved document set. This space is created at retrieval time and is characterised by the presence of search terms (i.e. it is query-relevant). Exploring it allows searchers to deeply examine search results and facilitates access to potentially useful information. Searchers can interact with *document representations* and follow *relevance paths* between these representations, generating evidence for the implicit models we evaluate. A similar granular approach has been shown to be effective in previous studies [16].

2.1 Document Representations

Documents are represented in the information space by their full-text and a variety of smaller, query-relevant representations, created at retrieval time. These include the document title and a four-sentence query-biased summary of the document [15]; a list of *top-ranking sentences* (TRS) extracted from the top thirty documents retrieved, scored in relation to the query, and; each summary sentence in the context it occurs in the document (i.e. with the preceding and following sentence). Each summary sentence and top-ranking sentence is regarded as a representation of the document. Since the full-text of documents can contain irrelevant information, shifting the focus of interaction to the query-relevant parts reduces the likelihood that erroneous terms will be selected by the implicit feedback models.

2.2 Relevance Paths

The six types of document representations described in Section 2.1 combine to form a *relevance path.* The further along a path a searcher travels the more relevant the information in the path is assumed to be. The paths can vary in length from one to six representations, and searchers can access the full-text of the document from any step in the path. *Relevance paths can startfrom top-ranking sentences or document titles.* Certain aspects of the path order are fixed e.g. the searcher must view a summary sentence before visiting that sentence in context. Figure 1 illustrates an example relevance path on an experimental search interface based on [16].

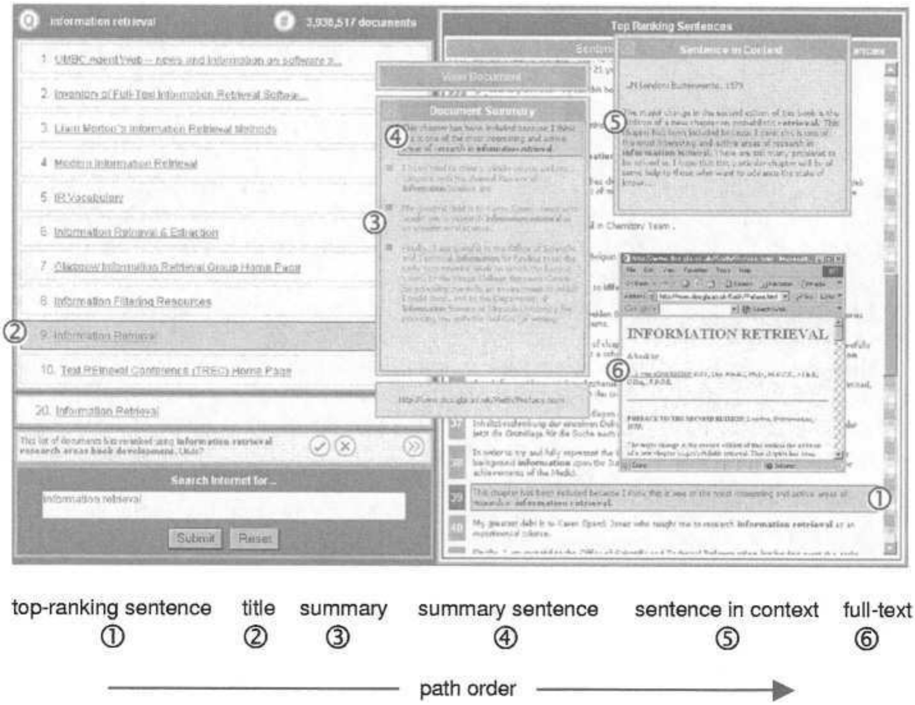


Fig. 1. The relevance path

Some representations of each document are fixed in content, i.e. the title and full-text of the document, whereas other representations, such as the summary, are dependent on the query and hence variable in content. Therefore, for each document, there may be many *potential* relevance paths. We use the distance travelled along the path and the particular representations viewed as evidence for the implicit models described in the next section.

3 Implicit Feedback Models

We developed six different implicit models that will be discussed in this section. The relevance assessments in *all* models are obtained implicitly, by interpreting a searcher's selection of one information object over others as an indication that this object is more relevant.

We encourage searchers to deeply examine the results of their search, following relevance paths and *exploring* the information space. All approaches use this exploration as a source of implicit evidence and choose the potentially relevant terms to expand the query. The presence of an information space allows certain models to retain some memory of searcher preferences and behaviour. This memory facilitates *learning* (i.e. the models learn over time what terms are relevant). The models presented learn in different ways, and in this section we describe each of them. All models, with the exception of the document-centric approach described in Section 3.3.1, use the document representations and relevance paths described in Section 2.

3.1 Binary Voting Model

The *binary voting model* [16] is a heuristic-based implicit feedback model. To identify potentially useful expansion terms the model allows each document representation to 'vote' for the terms it contains. When a term is present in a viewed representation it receives a 'vote', when it is not present it receives no vote. All terms are candidates in the voting process, and these votes accumulate across all viewed representations.

Different *types* of representation vary in length and may have a different indicative worth, typically measured based on representation length [1]. For example, a top-ranking sentence is shorter than a query-biased document summary (typically composed of four sentences) and is therefore less indicative of document content. To compensate for this, we use heuristic weights for the indicative worth of each type of representation. The weights used are 0.1 for title, 0.2 for top-ranking sentence, 0.3 for summary, 0.2 for summary sentence and 0.2 for sentence in context. These weights, based only on the *typical* length of a representation, ensure that the total score for a term in a relevance path is between 0 and 1 (inclusive).

The terms with the highest overall vote are those that are taken to best describe the information viewed by the searcher (i.e. those terms that are present most often across all representations) and can be used to approximate searcher interests.

3.2 Jeffrey's Conditioning Model

The next implicit model discussed uses *Jeffrey's rule of conditioning* [5] to revise the probability of term relevance in light of evidence gathered from searcher interaction. Jeffrey's conditioning captures the uncertain nature of implicit evidence, and is used since even after the passage of experience (i.e. following a relevance path) the model is still uncertain about the relevance of a term. The approach we use for this revision is based on that proposed by van Rijsbergen [12].

The binary voting model used a set of pre-defined heuristic weights for the indicativity of a path's constituent representations. In the Jeffrey's conditioning model we use various measures to describe the value, or worth, of the evidence a representation provides. We combine a confidence measure that uses the relative position of representations in the relevance path with a measure of indicativity based on the concepts in a representation. In this section we describe each of these measures, and how the Jeffrey's conditioning model weights potential expansion terms.

3.2.1 Path Weighting

For each path, we become more confident about the value of aged relevance information as we regress. In our approach we assign an exponentially increasing relevance profile to *aged* relevance. The representations that comprise the path are smaller than documents, the paths are generally short (i.e. no more than six representations) and the most recent document representation is not necessarily the most relevant.

The assumption we make is that the further we travel along a relevance path, the more certain we are about the relevance of the information towards the start of the path. As the viewing of the *next* representation is exploratory and *driven by curiosity as well as information need* we are cautious, and hence less confident about the value of this evidence. This confidence, *c,* is assigned *from the start of the path* to each representation *i,*

$$c_i = \frac{1}{2^i}, \text{ where } i \geq 1 \tag{1}$$

However, since across a whole path, the values of c_i do not sum to one, we must normalise and compute the confidence *c* for each representation *i* in a path of length *N* using,

$$c_i = \left(\frac{1}{2^i} + \frac{1}{N.2^N}\right), \text{ where } \sum_{i=1}^{N} c_i = 1 \text{ and } i \in \{1,2,\ldots,N\} \tag{2}$$

3.2.2 Indicativity and Quality of Evidence

In the previous section we described the confidence in the relevance of representations based on their position in the relevance path. The quality of evidence in a representation, or its *indicative worth,* can also affect how confident we are in the value of its content. In the binary voting model we use heuristics based on the *typical* length of document representations to measure indicativity. However, titles and top-ranking sentences, which may be very indicative of document content, are short and will have low indicativity scores if their typical length is the attribute used to score them.

In this approach, we use the non-stopword terms, or *concepts,* in a representation instead of representation length. We weight a term *t* in document *d* using its normalised term frequency [4], and the sum of all weights in a document is 1. The larger this value, the more often it occurs in the document, and the more representative of document content that term can be seen to be. To compute the indicativity index *I* for a representation *r* we sum the weight of a term in a document $w_{t,d}$ for all *unique* terms in *r,*

$$I_r = \sum_{t \in r} w_{t,d} \tag{3}$$

The I_r ranges between 0 and 1, is never 0, and is 1 only if the representation contains every unique term in the document. The indicativity measure is only incremented if there is a match between the unique terms in the document and those in the representation[1].

Relevance paths will contain representations of varying quality. We compute the *value* of the evidence in a representation by multiplying its indicativity by its confidence. Using these measures ensures that the worthwhile representations in each relevance path contribute most to the selection of potentially useful query expansion terms. In the next section we describe how such terms are chosen.

[1] This measure is similar to a *Hamming distance* [3], but uses term *weights,* rather than presence/absence.

3.2.3 Term Weighting

The Jeffrey's model assumes the existence of a *term space T*, a mutually exclusive set of all (non-stemmed, non-stopword) terms in the information space. Each term in T is independent and has an associated frequency in the information space. We define the probability that a term t is relevant based on a probability distribution P over T as,

$$P(t) = \frac{ntf(t)}{\sum_{t\in T} ntf(t)} \quad \textbf{where } \boldsymbol{ntf(t)} \textbf{ is the } \textbf{\textit{normalised term frequency}} \text{ [4]} \; \textbf{of term } \boldsymbol{t} \textbf{ in the term space } \boldsymbol{T} \tag{4}$$

To update this probability based on new evidence gathered from interaction we use *Jeffrey's Rule of Conditioning,* applied at the end of each relevance path. We consider this relevance path p as a new source of evidence to update the probability to say P'.

The viewing of a representation p_i creates new evidence for the terms in that representation. We use Jeffrey's rule of conditioning to update the probabilities based on this new evidence using the following formula,

$$P'(t) = \left[P(t=1 \mid p_i) \frac{P'(t=1)}{P(t=1)} + P(t=0 \mid p_i) \frac{P'(t=0)}{P(t=0)} \right] . P(t) \tag{5}$$

This estimation calculates the revised probability of relevance for a term t given a representation p_i, where $P(t = 1)$ is the probability of observing t, and $P(t = 0)$ the probability of not observing t. This updated probability reflects the 'passage of experience' and is similar to that described in [12].

A relevance path contains a number of representations. We update the probabilities after the traversal of a relevance path. The length of a relevance path ranges between 1 and 6 steps. We denote this length using N. When this length is greater than one we update the probabilities across this path. The probability of relevance of a term across a path of length N is denoted P_N and given through *successive updating,*

$$P_N(t) = \sum_{i=1}^{N-1} c_i . I_i . \left[\left(P_i(t=1 \mid p_i) \frac{P_{i+1}(t=1)}{P_i(t=1)} + P_i(t=0 \mid p_i) \frac{P_{i+1}(t=0)}{P_i(t=0)} \right) . P_i(t) \right] \tag{6}$$

where a representation at step i in the path p is denoted p_i. The confidence in the value of the representation is denoted c_i and I_i is the indicativity of the representation. In this equation, the order of the updating matters, so the order in which the searcher traverses the path also matters.

The actual revision of the probabilities will occur after each path. Once learned, the probabilities of relevance remain stable until the next revision (i.e. the next relevance path). Only terms in T that appear in the relevance path will have their probabilities revised *directly*[2].

3.3 WPQ-Based Models

In this section we present three implicit feedback models that use the popular *wpq* method [8] to rank terms for query expansion. This method has been shown to be effective and produce good results. The equation for *wpq* is shown below, where the typical values r_t = the number of seen relevant documents containing term t, n_t = the number of documents containing t, R = the number of seen relevant documents for query q, N= the number of documents in the collection.

[2] Based on the new evidence probabilities are redistributed to make the sum 1.

$$wpq_t = \log \frac{r_t/(R-r_t)}{(n_t-r_t)/(N-n_t-R+r_t)} \cdot \left(\frac{r_t}{R} - \frac{n_t-r_t}{N-R} \right) \quad (7)$$

The *wpq* method is based on probabilistic distributions of a term in relevant and non-relevant documents. As the values of r_t and R change during searcher interaction, the *wpq*-generated term weights also change. However, there is no retained memory of these term weights between iterations, and wpq_t is recomputed after each iteration. The *wpq* approaches learn what *information objects* are relevant but do not directly 'remember' the weights assigned to *terms*. This is unlike the Jeffrey's and binary voting models, which store and revise term weights for the entire search session.

3.3.1 WPQ Document Model

The *wpq document model* uses the full-text of documents, rather than granular representations or paths that link them. The *wpq* formula is applied to each document and expansion terms chosen from it. The values of R = the number of seen documents, r_t = the number of seen documents containing term *t*, N = the number of top-ranked documents and n_t = the number of top-ranked documents containing the term *t*. This approach is effectively a traditional explicit relevance feedback model, choosing one relevant document per iteration. This is a realistic model since implicit feedback is typically gathered sequentially (i.e. one relevance indication after another) and was included in the study to investigate the effects of using whole documents for such feedback.

3.3.2 WPQ Path Model

In the *wpq path model* the terms from each complete relevance path are pooled together and ranked based on their *wpq* score. We use the values R = the number of seen paths, r_t = the number of seen paths containing term *t*, N = the total number of paths generated from the top 30 retrieved documents, n_t = the number of generated paths that contain the term *t*. Since it uses terms in the *complete path* for query expansion, this model does not use any path weighting or indicativity measures. This model was chosen to investigate combining *wpq* and relevance paths for implicit feedback.

3.3.3 WPQ Ostensive Profile Model

The *wpq ostensive profile model* considers each representation in the relevance path separately, applying the *wpq* formula and ranking the terms each representation contains. This model adds a temporal dimension to relevance, assigning a within-path *ostensive relevance profile* [2] that suggests a recently viewed step in the relevance path is more indicative of the current information need than a previously viewed one. This differs from the Jeffrey's model, which assigns a reduced weight to most recently viewed step in the path. The *wpq* weights are normalised using such a profile. The model treats a relevance path a series of representations, and uses each representation separately for *wpq*. In this model the *wpq* formula uses the values R = the number of seen representations, r_t = the number of seen representations containing term *t*, N = the number of representations in top-ranked documents, n_t = the number of representations containing the term *t*. This model uses an ostensive relevance profile to enhance the *wpq path model* presented in the previous section.

3.4 Random Term Selection Model

The random term selection model assigns a random score between 0 and 1 to terms from viewed representations. At the end of each relevance path, the model ranks the terms based on these random scores and uses the top-scoring terms to expand the original query. This model does not use any path weighting or indicativity measures. This model is a baseline and was included to test the degree to which using any reasonable term-weighting approach affected the success of the implicit feedback. Also, since it did not retain any memory of important terms or information objects this model was also expected to experience no learning.

3.5 Summary

We have introduced a variety of implicit feedback models based on binary voting, Jeffrey's rule of conditioning, three using *wpq* query expansion and random term selection. In this study we compare these models based on the degree to which each improves search effectiveness and learns relevance. In the next section we describe the searcher simulation that tests these models.

4 Simulation-Based Evaluation Methodology

There has been no precedent set on how to best evaluate implicit feedback models. In this study we use a simulation-based evaluation methodology to benchmark such models and choose the best performing models for future studies with real searchers.

The simulation assumes the role of a searcher, browsing the results of an initial retrieval. The information content of the top-ranked documents in the first retrieved document set constitutes the information space that the searcher must explore. All interaction in this simulation is with this set (i.e. we never generate a new information space) and we assume that searchers will only view relevant information (i.e. only follow relevance paths from relevant documents).

4.1 System, Corpus, and Topics

We use the popular SMART search system [11] and index the San Jose Mercury News (SJMN 1991) document collection taken from the TREC initiative [13]. This collection comprises 90,257 documents, with an average 410.7 words per document (including document title), an average 55.6 relevant documents per topic and has been used successfully in previous experiments of this nature [9].

We used TREC topics 101-150 and took query from the short *title* field of the TREC topic description. For each query we use the top 30 documents to generate relevance paths for use in our simulation. Although the collection comes with a list of 50 topic (query) descriptions, we concentrate on those queries with relevant documents from which to generate relevance paths. We exclude those queries where there are no relevant documents in the top 30 documents retrieved and queries for which there were no relevant documents. We use 43 of the original 50 topics in our study.

4.2 Relevance Paths

Real searchers would typically follow a series of *related* relevance paths in a rational way, viewing only the most useful or interesting. In this study we try to simulate the searcher, but do not make such decisions. Instead, we select a set of paths from the large set of potential paths generated *at random* from top-ranked relevant documents.

Each relevant document has a number of possible relevance paths. In Table 1 we give all routes for all path types. Since we deal with granular representations of documents, we do not include the sixth and final *Document* step in these paths.

Table 1. Possible relevance path routes

TRS	Title	Summary	Summary Sentence	Sentence in Context	Total
4	1	1	4	1	**16**
4	1	1	4		**16**
4	1	1			**4**
4	1				**4**
4					**4**
	1	1	4	1	**4**
	1	1	4		**4**
	1	1			**1**
	1				**1**

For example, for viewing all five representations (first row of Table 1) there are $4\times1 \times 1\times 4 \times 1 = 16$ possible paths. The final column shows the total for each possible route. There are 54 possible relevance paths for each document. If all top 30 documents are relevant there are 1,620 (54 × 30) possible relevance paths.

In our study we use only a subset of these possible paths. The simulation assumes that searchers interact with relevant information, and not with every possible relevance path. Even though it was possible to use all paths for each query, different queries have different numbers of relevant top-ranked documents (and hence possible relevance paths). For the sake of comparability and consistency, we only use a subset of these paths, chosen randomly from all possible. The subset size is constant for all models.

4.3 Relevant Distributions and Correlation Coefficients

A good implicit feedback model should, given evidence from relevant documents, learn the distribution across the relevant document set. The model should train itself, and become attuned to searcher needs in the fewest possible iterations.

A relevant term space for each topic is created before any experiments are run. This space contains terms from all the relevant documents for that topic, ordered based on their probability of relevance for that topic, computed in the same way as Equation 4.

After each iteration we calculate the extent to which the term lists generated by the implicit model correlates with the relevant distribution. The simulation 'views' relevance paths from relevant documents and provides the models with the implicit relevance information they need to train themselves. We measure how well the models *learn* relevance based on how closely the term ordering they provide matches the term ordering in the relevant distribution.

To measure this we use two nonparametric correlation coefficients, *Spearman's rho* and *Kendall's tau.* These have equivalent underlying assumptions and statistical power, and both return a coefficient in the range [-1,1]. However, they have different interpretations; the Spearman accounts for the proportion of variability between *ranks* in the two lists, the Kendall represents the difference between the probability that the lists are in the same order versus the probability that the lists are in different orders. We used both coefficients to verify learning trends.

4.4 Evaluation Procedure

The simulation creates a set of relevance paths for all relevant documents in those top-ranked documents retrieved for each topic. It then follows a random-walk of *m* relevance paths, each path is regarded as a feedback *iteration* and *m* is chosen by the experimenter. After each iteration, we monitor the effect on search effectiveness and how closely the expansion terms generated by the model correlate with the term distribution across that topic's relevant documents. We use this correlation as a measure of how well the model learns the relevant term distribution and precision as a measure of search effectiveness.

The following procedure is used *for each topic with each model:*

i. use SMART to retrieve document set in response to query (i.e. topic title) using an *idf* weighting scheme

ii. identify relevant documents in the top 30 retrieved documents

iii. create a query-biased summary of all relevant documents from top 30 in parallel using the approach presented in [15]

iv. create and store all potential relevance paths for each relevant document (up to a maximum of 54 per document)

v. choose random set of *m* relevance paths (iterations) from those stored (using the Java[3] random number generator)

vi. for *each* of the *m* relevance paths:

    a. weight terms in path with chosen model
    b. monitor Kendall and Spearman by comparing order of terms with order in that relevant distribution for that topic
    c. choose top-ranked terms and use them to expand original query
    d. use new query to retrieve new set of documents
    e. compute new precision and recall values

[3] http://java.sun.com

To better represent a searcher exploring the information space, all simulated interaction was with the results of the first retrieval only. All subsequent retrievals were to test the effectiveness of the new queries and were not used to generate relevance paths. In the next section we describe our study.

4.5 Study

In our study we test how well each model learned relevance and generated queries that enhanced search effectiveness. We ran the simulation ten times for each implicit model, over all 43 'useable' topics. We added six terms to the query, this was done without any prior knowledge of the effectiveness of adding this number of terms to queries for this collection. We set $m = 20$ and hence *each run comprised 20 iterations* (i.e. relevance paths or documents). We recorded correlation coefficients and measures of search effectiveness at iterations 1, 2, 5, 10 and 20. Using these iterations allowed us to monitor performance at different points in the search. In the document-centric approach each *document* is an iteration. Therefore, in this model, it was only possible to have as many iterations as there were relevant top-ranked documents.

5 Results

The study was conducted to evaluate a variety of implicit feedback models using searcher simulations. In this section we present results of our study. In particular we focus on results concerning search effectiveness and relevance learning. We use the terms *bvm, jeff, wpq.doc, wpq.path, wpq.ost* and *ran* to refer the binary voting, Jeffrey's, wpq document, wpq path, wpq ostensive and random models respectively.

5.1 Search Effectiveness

We measure search effectiveness for each of our implicit models through their effects on precision[4]. Figure 2 shows the 11pt precision values for each model across all iterations. As the figure illustrate, all models increased precision as the number of iterations increases.

Figure 2 presents the actual precision values across all 20 iterations. The Jeffrey's and binary voting models outperform the other implicit feedback models, with large increases inside the first five iterations. Both models are quick to respond to implicit relevance information, with the largest marginal increases (change from one iteration to the next) coming in the first iteration. The other models do not perform as well, but steadily increase until around 10 iterations where precision levels out.

Table 2 illustrates the marginal difference more clearly than Figure 2, showing the percentage change overall and the marginal percentage change at each iteration.

As Table 2 shows the largest increases in precision overall and marginally come from the binary voting model and the Jeffrey's model. Although after 20 iterations the

[4] Both precision and recall were improved by the implicit models. However, since we only consider the top-30 documents the effects on precision are of more interest in this study.

marginal effects of all models appear slight. The random model performs poorly, although still leads to small overall increases in precision over the baseline. Even though the *random model* assigned each term a random score, the paths selected by the simulation were still query-relevant. Our results show that choosing terms randomly from relevance paths can help improve short queries to a small degree.

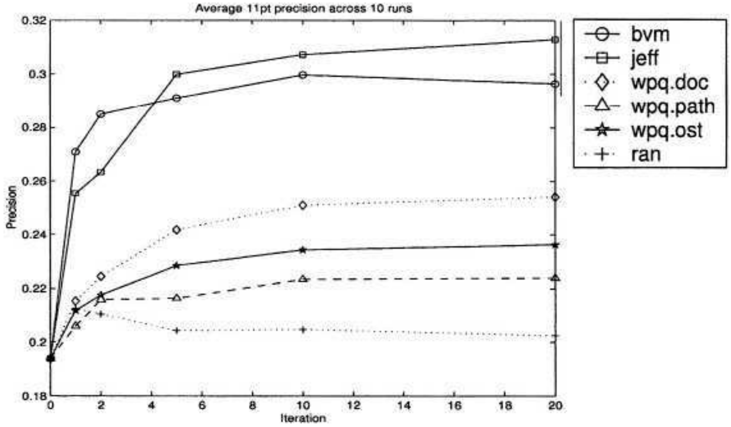


Fig. 2. Average precision across 20 feedback iterations

Table 2. Percentage change in precision per iteration. Overall change in first column, marginal change in second shaded column. Highest percentage in each column in bold

	Iterations									
Model	1		2		5		10		20	
bvm	**28.4**	–	**31.9**	**4.9**	33.4	2.9	35.3	2.9	34.6	–1.1
jeff	24.1	–	26.4	3.0	**35.3**	**12.2**	**36.9**	2.4	**38**	**1.8**
wpq.doc	10	–	13.6	4.1	19.8	7.1	22.8	**3.7**	23.7	1.2
wpq.path	5.8	–	10.2	4.6	10.4	0.2	13.2	3.2	13.4	0.2
wpq.ost	8.5	–	10.9	2.6	17.2	4.8	17.2	2.5	18	0.9
ran	8.8	–	7.9	–1.1	5	–3.1	5.3	0.2	4.2	–1.1

The *wpq*-based models appeared to follow a similar trend. At each iteration we carried out a one-way repeated measures ANOVA to compare all three *wpq*-based models and *t*-tests for pair-wise comparisons where appropriate. During the first two iterations, there were no significant differences (iteration 1: $F_{2,27} = 2.258$, $p = 0.12$, iteration 2: $F_{2,27} = 1.803$, $p = 0.18$) between the *wpq* models tested. ANOVAs across iterations 5, 10 and 20 suggested there were significant differences in precision between the three *wpq*-models. A series of *t*-tests revealed the *wpq document model* performed significantly better than both path-based *wpq* models (ostensive-path and path) for iterations 5, 10 and 20 ($p < 0.05$). We could therefore posit that perhaps the relevance paths were not of sufficient size and did not contain a sufficient mixture of terms from which *wpq* could choose candidates for query expansion.

5.2 Relevance Learning

We measured how well the implicit models trained themselves when given relevance information by the simulation by using the degree of correlation between the ordered list of terms in the topic's relevant distribution and the ordered list of terms chosen by the implicit model. Figure 3 shows the average Spearman (a) and Kendall (b) correlation coefficients across all 43 topics.

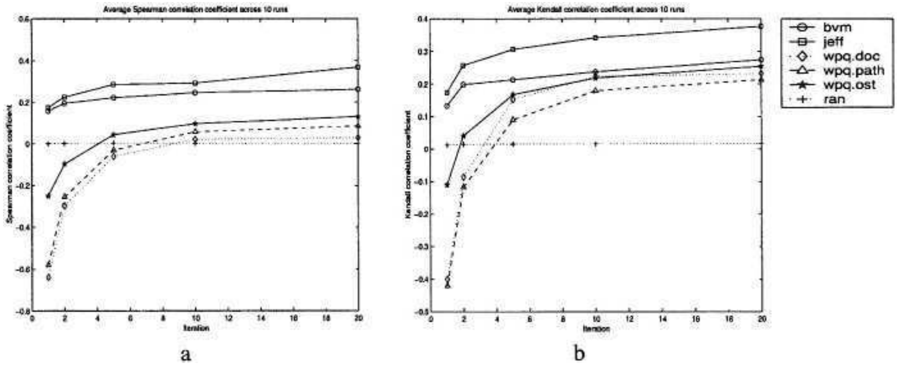


Fig. 3. Correlation coefficients; a. Spearman b. Kendall

Both coefficients follow similar trends for all implicit models. Again the Jeffrey's and binary voting model learn at a faster rate, with the Jeffrey's performing best. The random model returns a coefficient value close to zero with both coefficients. In both cases a value of zero implies no correlation between the two lists, and this was to be expected if the model randomly ordered the term list. For all other models the coefficients tends to one, implying that the models were *learning* the relevant distribution from the given relevance information. Both the Jeffrey's model and the binary voting model obtain high degrees of correlation after the first iteration, whereas the *wpq* models need more *training* to reach a level where the terms they recommend appear to match those in the relevant distribution.

6 Discussion

The implicit feedback models evaluated in this paper *all* increased search effectiveness through query expansion. However, two models performed particularly well; that based on Jeffrey's conditioning and the binary voting model. Both models improved precision and developed lists of terms that were closely correlated to those of the relevant distribution.

Initially the Jeffrey's does not perform as well as the binary voting model. However, after five paths it creates more effective queries and from then on performs increasingly better than it. The Jeffrey's model uses prior evidence that is independent of the searcher's interaction. Initial decisions are made based on this prior evidence, and for the first few iterations it is reasonable to assume that this evidence still plays a part in term selection. However, as more evidence is gathered from searcher interaction the terms selected by Jeffrey's conditioning improve.

An advantage of the binary voting model, and perhaps why it performs well in the initial stages is that it does not rely on any prior evidence, selecting terms based only on the representations viewed by the searcher. However, the lists of potential terms offered stagnates after 10 paths, since in the binary voting model the effect of the scoring is cumulative, the high-scoring, high-occurrence terms, obtain a higher score after only a few initial paths and cannot be succeeded by lower-ranked terms in later paths. This often means that the same query is presented in iterations 10 and 20.

The implicit feedback models learned relevance from the evidence provided to them by the simulation. This form of *reinforcement learning* [7], where the model was repeatedly shown examples of relevant information, allowed us to test how well each model trained itself to recognise relevance. From the six models tested, our findings showed that the Jeffrey's and binary voting models learned at the fastest rate. In the first few iterations those models based on *wpq* performed poorly, suggesting that these models need more training to reach an acceptable level of relevance recognition and that the Jeffrey's and binary voting models make a more efficient use of the relevance information presented to them.

We used linear regression and compared the *rate of learning* against *precision* for each of the six implicit feedback models. The results showed that for all models, the rate of learning (i.e. *Spearman's rho* and *Kendall's tau*) followed the same trend as precision (*all* $r^2 \geq .8856$ and *all* $T_{38} \geq 6.48, p \leq .05$). The rate in which the models learn relevance appears to match the rate in which they are able to improve search effectiveness.

For almost all iterations on all models, the marginal increases in precision and correlation reduce as more relevant information is presented. The models appear to reach a point of saturation at around 10 paths, where the benefits of showing 10 more paths (i.e. going to iteration 20) are only very slight and are perhaps outweighed by the costs of further interaction. It is perhaps at this point where searcher needs would be best served with a new injection of different information or explicit searcher involvement.

The results appear to be collection-independent. We re-ran the same experiment using the Wall Street Journal 1990-1992 collection instead of SJMN 1991. The findings mirrored those obtained in this study.

In the absence of a proper methodology for evaluating interactive retrieval approaches we introduced a novel simulation-based evaluation strategy. In this scheme we simulate searcher actions through a relevant set of document representations. However, a potential drawback of the searcher simulation proposed in this paper is that it does not consider the intentionality in interaction. A real searcher will view a series of information objects in rational way, depending on their information need. The simulation chooses paths *at random* from the top-ranked documents, and uses these paths to simulate interaction. At present the information need persists at the relevant document level (i.e. we choose paths from relevant documents), we posit that if the simulation catered for persistence in the interaction (i.e. relevance paths were traversed rationally) then the increases in search effectiveness and relevance learning would perhaps be even higher than those obtained.

7 Conclusions

In this paper we used searcher simulations to evaluate of a variety of implicit feedback models. The models under test are ostensive in nature and use the exploration of the information space and the viewing of information objects as an indication of relevance. We tested six models in total, all using an ostensive paradigm but each employing a different term selection stratagem.

We introduced implicit models based on Jeffrey's rule of conditioning, binary voting and three that use the popular *wpq* query expansion approach. The simulated approach used to test the model assumes the role of a searcher 'viewing' relevant documents and relevance paths between granular representations of documents. The simulation passes the information it viewed to the implicit models, which use this evidence of relevance to select terms to best describe this information. We investigated the degree to which each of the models improved search effectiveness and learned relevance. From the six models tested, the Jeffrey's model provided the highest levels of precision and the highest rate of learning.

The burden of explicitly providing relevance information in traditional relevance feedback systems makes implicit feedback an appealing alternative. Simulation experiments are a reasonable way to test the worth of implicit models such as those presented in this paper. These tests can ensure that only the most effective implicit models are chosen as potential substitutes for explicit RF in interactive information seeking environments. Implicit systems using the Jeffrey's model are under development.

Acknowledgements. The work reported is funded by the UK EPSRC, grant number GR/R74642/01.

References

1. Barry, C.L. 'Document Representations and Clues to Document Relevance'. *Journal of the American Societyfor Information Science.* 49. 14, 1293-1303. 1998.
2. Campbell, I. and van Rijsbergen, C.J. 'The ostensive model of developing information needs'. *Proceedings of the 3rd CoLIS Conference,* 251-268. 1996.
3. Hamming, R.W. 'Error-Detecting and Error-Correcting Codes', *Bell Systems Technical Journal.* 29. pp 147-160. 1950.
4. Harman, D. 'An Experimental Study of the Factors Important in Document Ranking'. In *Proceedings of the 9th ACM SIGIR Conference,* 186-193. 1986.
5. Jeffrey, R.C. *The Logic ofDecision,* 2nd edition. University of Chicago Press. 1983.
6. Lam, W., Mukhopadhyay, S., Mostafa, J., and Palakal, M. 'Detection of Shifts in User Interests for Personalised Information Filtering'. *Proceedings of the 18th ACM SIGIR Conference,* 317-325. 1996.
7. Mitchell, T.M. *Machine Learning.* McGraw-Hill. 1997.
8. Robertson, S.E. 'On term selection for query expansion'. *Journal of Documentation.* 46. 4, 359-364. 1990.
9. Ruthven, I. 'Re-examining the Potential Effectiveness of Interactive Query Expansion'. *Proceedings of the 26th ACM SIGIR Conference,* 213-220. 2003.
10. Salton, G. (Ed.). *The SMART Retrieval System.* Prentice-Hall. 1971.

11. Salton, G. and Buckley, C. ‘Improving retrieval performance by relevance feedback’. *Journal of the American Society for Information Science.* 41. 4. pp 288-297. 1990.
12. van Rijsbergen, C.J. ‘Probabilistic Retrieval Revisited’. *The Computer Journal.* 35. 3, 291-298. 1992.
13. Voorhees, E.H. and Harman, D. ‘Overview of the sixth text retrieval conference (TREC-6)’. *Information Processing and Management.* 36. 1, 3-35. 2000.
14. White, R.W., Jose, J.M. and Ruthven, I. The use of implicit evidence for relevance feedback in Web retrieval. *Proceedings of24th ECIR Conference,* 93-109. 2002.
15. White, R.W., Jose, J.M. and Ruthven, I. A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Information Processing and Management.* 39. 5, 707-733. 2003.
16. White, R.W., Jose, J.M. and Ruthven, I. An Approach for Implicitly Detecting Information Needs. *Proceedings of 12th CIKM Conference,* 504-508. 2003.

Cross-Language Information Retrieval Using EuroWordNet and Word Sense Disambiguation

Paul Clough and Mark Stevenson

University of Sheffield
Regent Court, 211 Portobello Street,
Sheffield, S1 4DP
United Kingdom
p.d.clough@sheffield.ac.uk,marks@dcs.shef.ac.uk

Abstract. One of the aims of EuroWordNet (EWN) was to provide a resource for Cross-Language Information Retrieval (CLIR). In this paper we present experiments which test the usefulness of EWN for this purpose via a formal evaluation using the Spanish queries from the TREC6 CLIR test set. All CLIR systems using bilingual dictionaries must find a way of dealing with multiple translations and we employ a Word Sense Disambiguation (WSD) algorithm for this purpose. It was found that this algorithm achieved only around 50% correct disambiguation when compared with manual judgement, however, retrieval performance using the senses it returned was 90% of that recorded using manually disambiguated queries.

1 Introduction

Cross-language information retrieval (CLIR) is the process of providing queries in one language and returning documents relevant to that query which are written in a different language. This is useful in cases when the user has enough knowledge of the language in which the documents are returned to understand them but does not possess the linguistic skill to formulate useful queries in that language. An example is e-commerce where a consumer may be interested in purchasing some computer equipment from another country but does not know how to describe what they want in the relevant language.

A popular approach to CLIR is to translate the query into the language of the documents being retrieved. Methods involving the use of machine translation, parallel corpora and machine readable bilingual dictionaries have all been tested, each with varying degrees of success [1,2,3]. One of the simplest and most effective methods for query translation is to perform dictionary lookup based on a bilingual dictionary. However, the mapping between words in different languages is not one-to-one, for example the English word “bank” is translated to French as “banque” when it is used in the ‘financial institution’ sense but as “rive” when it means ‘edge of river’. Choosing the correct translation is important for retrieval since French documents about finance are far more likely to contain the

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 327–337, 2004.

word “banque” than “rive”. A CLIR system which employs a bilingual dictionary must find a way of coping with this translation ambiguity.

The process of identifying the meanings of words in text is known as word sense disambiguation (WSD) and has been extensively studied in language processing. WSD is normally carried out by selecting the appropriate sense for a context from a lexical resource such as a dictionary or thesaurus but for CLIR it is more appropriate to consider the set of senses as the possible translations of a term between the source and target languages. For example, in an English-to-French CLIR system the word “bank” would have (at least) two possible senses (the translations “banque” and “rive”). By considering the problem of translation selection as a form of WSD allows us to make use of the extensive research which has been carried out in that area.

EuroWordNet (EWN) [4] is a lexical database which contains possible translations of words between several European languages and was designed for use in CLIR [5]. In this paper we use EWN to provide a set of possible translations for queries into the target language. A WSD algorithm is used to choose the correct meaning from the possibilities listed. The translated queries are then sent to a retrieval engine.

Section 2 provides a description of EuroWordNet, focussing on the most relevant to the work described here. Section 3 describes the WSD algorithm we use to resolve ambiguity in the retrieval queries. In Section 4 we describe the experiments which were used to determine the improvement in performance which may be gained from using WSD for CLIR the results of which are presented in Section 5. Section 6 describes an evaluation of the WSD algorithm used. The implications and conclusions which can be drawn from this work are presented in Sections 7 and 8.

2 EuroWordNet

WordNet [6] is a semantic database for English which was constructed at Princeton University. It was originally designed to model the psycholinguistic construction of the human mental lexicon but quickly proved to be an extremely popular resource in language processing [7]. The basic building block of WordNet is the synset (SYNonym SET): a group of words with closely related meanings. Words with more than one meaning belong to multiple synsets. For example, the noun “car” has 5 different meanings (senses), including a railway car, elevator car, automobile and cable car. Sense two of car (`car#2`) has synset members: `[car, motorcar, machine, auto, automobile]`. The first entry in a synset is understood to represent the most frequent term which refers to the use of a particular synset and is known as the *head* of the synset. For example, in the previous example, “car” is the head.

WordNet synsets are connected through a series of relations the most important of which are hyponymy (`IS_A`) and hyperonymy (`KIND_OF`). The hyponomy relation is used for generalisations of the concepts in the synset while the hyperonymy relation lists specialisations. For example, the hyperonym of the synset containing `car#2` is `[motor vehicle, automotive vehicle]` and a hyponym `[cab, taxi, hack, taxicab]`. The hyponymy and hyperonymy relations serve to organise the lexicon into a hierarchical structure with separate hierarchies for nouns, verbs, adjectives and adverbs. The WordNet structure is essentially tree-like since synsets often have multiple hyponyms, however it is not a tree in a strict sense since synsets usually have a unique hyponym but a small number have more.

EuroWordNet [4] is a multi-lingual lexical resource created by extending the original WordNet to include coverage of Dutch, Italian, Spanish, French, German, Czech and Estonian. Through a number of initiatives it has been extended to cover Basque, Portugese and Swedish.[1] One of the objectives in the construction of EuroWordNet was to provide an integrated resource which could be used for CLIR [8]. This has been achieved by linking together the various language-specific WordNets through a language-independent Inter-Lingual Index (ILI). In practise the ILI is closely based on a version of the original English WordNet. A set of equivalence relations are used to identify concepts which are equivalent across different languages [4]. For practical reasons these relations were not implemented between each language pair. Instead each language provides a map between itself and the ILI. So, for example, to translate a EuroWordNet concept from Spanish to German it would be necessary to translate that concept from Spanish to the equivalent ILI concept and from that to appropriate concept in the German WordNet.

3 Word Sense Disambiguation

One of the main challenges in using a resource such as EWN is discovering which of the synsets are appropriate for a particular use of a word. For example, the `car#2` synset of “car” would be relevant to a query about automobiles while the alternatives, which include “elevator car” and “railroad car” would not. In order to do this we adapted a WSD algorithm for WordNet originally developed by Resnik [9]. The algorithm is designed to take a set of nouns as context and determine the meaning of each which is most appropriate given the rest of the nouns in the set. This algorithm was thought to be suitable for disambiguating the nouns in retrieval queries.

The algorithm is fully described in [9] and we shall provide only a brief description here. The algorithm makes use of the fact that WordNet synsets are organised into a hierarchy with more general concepts at the top and more specific ones below them. So, for example, `motor vehicle` is less informative than `taxi`. A numerical value is computed for each synset in the hierarchy by counting the frequency of occurrence of its members in a large corpus[2]. This value is dubbed the *Information Content* and is calculated as $\textit{Information Content}(\textit{synset}) = -\log \Pr(\textit{synset})$.

[1] `http://www.globalwordnet.org`

[2] We used the British National Corpus which contains 100 million words.

The similarity of two synsets can be estimated as the information content value of most specific synset (i.e. the one with the highest information content) which subsumes both. So, for example, the words `nurse` and `doctor` are subsumed by `health professional`, `person` and other synsets. The most specific of these is `health professional`. The information content of the relevant synset containing `health profession` then be chosen as the similarity of the words `nurse` and `doctor`. By extension of this idea, sets of nouns can be disambiguated by choosing the synsets which return the highest possible total information content value. For each sense a value is returned indicating the likelihood of the sense being the appropriate one given the group of nouns.

4 Experimental Setup

4.1 Test Collection

Evaluation was carried out using past results from the cross-lingual track of TREC6 [10]. We used only TREC6 runs that retrieved from an English language collection, which was the 242,918 documents of the Associated Press (AP), 1988 to 1990. NIST supplied 25 English CLIR topics, although four of these (topics 3, 8, 15 and 25) were not supplied with any relevance judgements and were not used for this evaluation.

The topics were translated into four languages (Spanish, German, French and Dutch) by native speakers who attempted to produce suitable queries from the English version. For this evaluation the Spanish queries were used to evaluate the cross-lingual retrieval and the English queries to provide a monolingual baseline. Spanish was chosen since it provides the most complete and accurate translation resource from the EWN languages. In addition the EWN entries for Spanish tend to have more senses than several of the other languages and is therefore a language for which WSD is likely to be beneficial.

In order to evaluate the contribution of the WSD algorithm and EWN separately the English and Spanish queries were manually disambiguated by the authors. The possible synsets were identified for each query (for the Spanish queries these were mapped from the Spanish synsets onto the equivalent English ones which would be used for retrieval). A single sense from this set was then chosen for each term in the query.

4.2 CLIR System

Our CLIR system employs 3 stages: term identification, term translation and document retrieval. The term identification phase aims to find the nouns and proper names in the query. The XEROX part of speech tagger [11] is used to identify nouns in the queries. Those are then lemmatised and all potential synsets identified in EWN.[3] For English queries this set of possible synsets were passed

[3] For these experiments the Spanish lemmatisation was manually verified and altered when appropriate. This manual intervention could be omitted given an accurate Spanish lemmatiser.

onto the WSD algorithm to allow the appropriate one to be chosen. Once this has been identified the terms it contains are added to the final query. (In the next Section we describe experiments in which different synset elements are used as query terms.) For Spanish queries the EWN Inter-Lingual-Index [4] was used to identify the set of English WordNet synsets for each term which is equivalent to to the set of possible translations. For each word this set of synsets was considered to be the set of possible senses and passed to the WSD algorithm which chooses the most appropriate. Non-translatable terms were included in the final translated query because these often include proper names which tend to be good topic discriminators.

Document retrieval was carried out using our own implementation of a probabilistic search engine based on the BM25 similarity measure (see, e.g. [12]). The BM25 function estimates term frequency as Poisson in distribution, and takes into account inverse document frequency and document length. Based on this weighting function, queries are matched to documents using a similarity measure based upon term co-occurrence. Any document containing at least one or more terms from the query is retrieved from the index and a similarity score computed for that document:query pair. Documents containing any number of query terms are retrieved (creating an OR'ing effect) and ranked in descending order of similarity under the assumption that those nearer the top of the ranked list are more relevant to the query than those nearer the bottom.

4.3 Evaluation Method

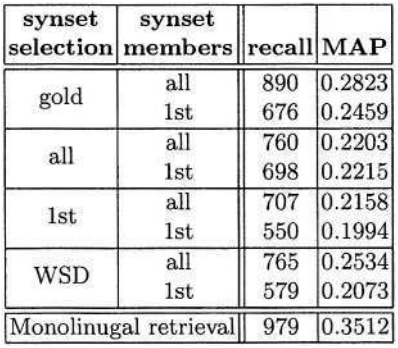
We experimented with various methods for selecting synsets from the query terms: all synsets, the first synset and the synset selected by the WSD algorithm. It is worth mentioning here that WordNet synsets are ordered by frequency of occurrence in text and consequently the first synset represents the most likely prior sense and a reasonable baseline against which the influence of WSD can be measured. We also varied the number of synset members selected: either the headword (first member of the synset), or all synset terms. In the case of all synset terms, we selected only distinct terms between different synsets for the same word (note this still allows the same word to be repeated within a topic). This was done to reduce the effects of term frequency on retrieval, thereby making it harder to determine how retrieval effectiveness is affected by WSD alone. Preliminary experiments showed retrieval to be higher using distinct words alone. We also experimented with longer queries composed of the TREC6 title and description fields, as well as shorter queries based on the title only to compare the effects of query length with WSD.

Retrieval effectiveness is measured using the `trec_eval` program as supplied by NIST. With this program and the set of relevance documents as supplied with the TREC6 topics, we are able to determine how many relevant documents are returned in the top 1000 rank positions, and the position at which they occur. We use two measures of retrieval effectiveness computed across all 25 topics. The first is *recall* which measures the number of relevant documents retrieved. The second measure, *mean uninterpolated average precision* (MAP), is calculated as the average precision figures obtained after each new relevant document is seen [13].

5 CLIR Evaluation

The results of cross-lingual retrieval can be placed in context by comparing them against those from the monolingual retrieval using the English version of the title and description as the query. (EuroWordNet was not used here and no query expansion was carried out.) It was found that 979 documents were recalled with a MAP score of 0.3512.

Table 1. Results for Spanish retrieval with title and description



synset selection	synset members	recall	MAP
gold	all	890	0.2823
	1st	676	0.2459
all	all	760	0.2203
	1st	698	0.2215
1st	all	707	0.2158
	1st	550	0.1994
WSD	all	765	0.2534
	1st	579	0.2073
Monolinugal retrieval		979	0.3512

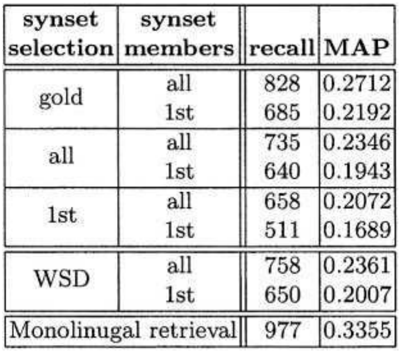
Table 1 shows retrieval results after translating the title and description. The first column ("synset selection") lists the methods used to choose the EWN synset from the set of possibilities. "gold" is the manually chosen sense, "all" and "1st" are the two baselines of choosing all possible synsets and the first while "auto" is the senses chosen by the WSD algorithm. The next column ("synset members") lists the synset members which are chosen for query expansion, either all synset members or the first one.

The best retrieval scores for manually disambiguated queries is recorded when all synset members are used in the query expansion which yields a MAP score of 0.2823 (see Table 1 row "gold", "all"). This is around 80% of the monolingual retrieval score of 0.3512. When WSD is applied the highest MAP score of 0.2534 is achieved when all synset members are selected (Table 1 row "WSD", "all"). This represents 72% of the MAP score from monolingual retrieval and 90% of the best score derived from the manually disambiguated queries.

In the majority of cases choosing all synset members leads to a noticeably higher MAP score than retrieval using the first synset member. This is probably because the greater number of query terms gives the retrieval engine a better chance of finding the relevant document. The exception is when all synsets have been selected (see Table 1). In this case the retrieval engine already has a large number of query terms through the combination of the first member from all synsets and adding more makes only a slight difference to retrieval performance.

When translating queries, it would appear that using Resnik's algorithm to disambiguate query terms improves retrieval performance when compared against choosing all possible senses or the first (most likely) senses to disambiguate.

Table 2. Results for Spanish retrieval with title only



synset selection	synset members	recall	MAP
gold	all	828	0.2712
	1st	685	0.2192
all	all	735	0.2346
	1st	640	0.1943
1st	all	658	0.2072
	1st	511	0.1689
WSD	all	758	0.2361
	1st	650	0.2007
Monolinugal retrieval		977	0.3355

The experiments were repeated, this time using just the title from the TREC query which represents a shorter query. The results from these experiments are shown in Table 2. As expected the baseline score obtained from the monolingual retrieval is lower than when both the title and description are used in the query. The manually annotated queries produces the highest MAP of 0.2712 (81% of monolingual). When the WSD algorithm is used the highest MAP is also recorded when all synset members were chosen. This score was 0.2361 (70% of monolingual). However, when the shorter queries are used the difference between WSD and the two naive approaches (choosing the most frequent sense and choosing all senses) is much smaller. This is probably because the reduced amount of context makes it difficult for the WSD algorithm to make a decision and it often returns all senses.

Table 2 also shows that choosing all synset members is a more effective strategy than choosing just the first member. We already noted this with reference to the results form the longer queries (Table 1) although the difference is more pronounced than when the longer queries were used. In fact it can be seen that when the short queries are used choosing all members for each possible synset (i.e. no disambiguation whatsoever) scores higher than choosing just the first member of the manually selected best sense. This shows that these shorter queries benefit far more from greater query expansion and that even correct meanings which are not expanded much do not provide enough information for correct retrieval.

6 Evaluation of WSD

It is important to measure the effectiveness of the WSD more directly than examining CLIR results. Others, such as [14] and [15], have found that WSD only has a positive effect on monolingual retrieval when the disambiguation is accurate. The manually disambiguated queries were used as a gold-standard against which the WSD algorithm we used could be evaluated. Two measures of agreement were computed: strict and relaxed. Assume that a word, w, has n senses denoted as $senses(w)(= w_1, w_2, ...w_n)$ and that one of these senses, w_{corr}(where $1 \leq corr \leq n$), was identified as correct by the human annotators. The WSD algorithm chooses a set of m senses, $wsd(w)$, where $1 \leq m \leq n$. The strict evaluation score for w takes into account the number of senses assigned by the WSD algorithm and if $w_{corr} \in wsd(w)$ the word is scored as $\frac{1}{m}$ (and 0 if $w_{corr} \notin wsd(w)$). The relaxed score is a simple measure of whether the WSD identified the correct senses regardless of the total it assigned and is scored as 1 if $w_{corr} \in wsd(w)$. The WSD accuracy for an entire query is calculated as the mean score for each term it contains.

The two evaluation metrics have quite different interpretations. The strict evaluation measures the degree to which the senses identified by the WSD algorithm match those identified by the human annotators. The relaxed score can be interpreted as the ratio of query words in which the sense identified as correct was not ruled out by the WSD algorithm. In fact simply returning all possible senses for a word would guarantee a score of 1 for the relaxed evaluation, although the score for the strict evaluation would probably be very low. Since it is important not to discard the correct sense for retrieval purposes the relaxed evaluation may be more relevant for this task.

Table 3. Results of WSD algorithm and first sense baseline compared against manually annotated queries

		Score	
Language	Method	Strict	Relaxed
English	WSD	0.410	0.546
	1st synset	0.474	
Spanish	WSD	0.441	0.550
	1st synset	0.482	

Table 3 shows the results of the evaluation of the WSD algorithm and baseline method of choosing the first sense against the manually annotated text for both the Spanish and English queries. The baseline scores are identical for each metric since it assigns exactly one sense for each word (the first) and the two metrics only return different scores when the technique assigns more than one sense.

We can see that the evaluation is similar across both languages. The baseline method actually outperforms automatic WSD according to the strict evaluation measure but scores less than it when the relaxed measure is used. We can also see that neither of the approaches are particularly accurate and often rule out the sense that was marked as correct by the human annotator.

However the results from the cross-language retrieval experiments earlier in this section show that there is generally an improvement in retrieval performance when the WSD algorithm is used. This implies that the relaxed evaluation may be a more appropriate way to judge the usefulness of a WSD algorithm for this task. This idea has some intuitive plausibility as it seems likely that for retrieval performance it is less important to identify the sense which was marked correct by an annotator than to try not to remove the senses which are useful for retrieval. It should also be borne in mind that the human annotation task was a forced choice in which the annotator had to choose exactly one sense for each ambiguous query term. In some cases it was very difficult to choose between some of the senses and there were cases where none of the EWN synsets seemed completely appropriate. On the other hand our WSD algorithm tended to choose several senses when there was insufficient contextual evidence to decide on the correct sense.

7 Discussion

The WSD algorithm's approach of only choosing senses when there is sufficient evidence suits this task well. However, the WSD results also highlight a serious limitation of EWN for CLIR. EWN's semantics are based on ontological semantics using the hyponymy relationship. That is, the EWN synset hierarchy contains information about the type of thing something is. So, for example, it tells us that "car" is a type of "motor vehicle". However, many types of useful semantic information are missing. One example is discourse and topic information. For example, "tennis player" (a hyponym of person) is not closely related to "racket", "balls" or "net" (hyponyms of artifact). Motivated by this example, Fellbaum [7] dubbed this the "tennis problem". This information is potentially valuable for retrieval where one aim is to identify terms which model the topic of the query. Others, including [1], [3] and [16], have used word co-occurrence statistics to identify the most likely translations and this could be considered a form of translation. This approach seems promising for CLIR since it returns words which occur together in text and these are likely to be topically related. This approach has potential to be developed into a WSD algorithm which could be applied to EWN.

There has been some disagreement over the usefulness of WSD for monolingual retrieval (see, for example, [15] and [17]). In particular [14] and [15] showed that WSD had to be accurate to be useful for monolingual retrieval. However, the results presented here imply that this is not the case for CLIR since the WSD methods were hindered by a lack of context and were not particularly accurate. The reason for this difference may be that retrieval algorithms actually perform a similar purpose to WSD algorithms in the sense that they attempt to identify instances of words being used with the relevant meanings. WSD algorithms therefore need to be accurate to provide any improvement. The situation is different for CLIR where identifying the correct translation of words in the query is unavoidable. This can only be carried out using some disambiguation method and the results presented here suggest that some disambiguation is better than none for CLIR.

8 Conclusions and Future Work

The results presented in this paper show that WSD is useful when CLIR was being carried out using EWN. The WSD algorithm used was not highly accurate on this particular task however it was able to outperformed two simple baselines and did not appear to adversely effect the retrieval results.

In future work we plan to experiment with different languages which are supported by EWN to test whether the differences in lexical coverage of the various EWNs have any effect on retrieval performance. We previously mentioned (Section 2) that EWN is designed around the English WordNet. In addition we plan to experiment with CLIR using two languages other than English to test whether this choice of architecture has any effect on retrieval performance. One of the authors has already shown that combining WSD algorithms can be a useful way of improving their effectiveness for ontology construction [18]. We plan to test whether similar techniques could be employed to improve the automatic disambiguation of queries.

Acknowledgments. The authors are grateful for advice from Mark Sanderson and Wim Peters of Sheffield University. The work described here was supported by the EPSRC-funded Eurovision project at Sheffield University (GR/R56778/01).

References

1. Ballesteros, L., Croft, W.: Resolving ambiguity for cross-language retrieval. In: Research and Development in Information Retrieval. (1998) 64–71
2. Jang, M., Myaeng, S., Park, S.: Using mutual information to resolve query translation ambiguities and query term weighting. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99), College Park, MA (1999) 223–229
3. Gao, J., Nie, J., He, H., Chen, W., Zhou, M.: Resolving query translation ambiguity using a decaying co-occurence model and syntactic dependence relations. In: Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retreival, Tampere, Finland (2002) 183–190
4. Vossen, P.: Introduction to EuroWordNet. Computers and the Humanities **32** (1998) 73–89 Special Issue on EuroWordNet.
5. Gilarranz, J., Gonzalo, J., Verdejo, F.: Language-independent text retireval with the EuroWordNet Multilingual Semantic Database. In: Proceedings of the Second Workshop on Multilinguality in the Software Industry: the AI contribution, Nagoya, Japan (1997) 9–16

6. Miller, G.: WordNet: An on-line lexical database. International Journal of Lexicography **3** (1990) 235–312
7. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database and some of its Applications. MIT Press, Cambridge, MA (1998)
8. Gilarranz, J., Gonzalo, J., Verdejo, F.: Language-independent text retrieval with the EuroWordNet Multilingual Semantic Database. In: Proceedings of the Second Workshop on Multilinguality in the Software Industry: the AI contribution at the Fifteenth International Joint Conference on Artificial Intelligence, Nagoya, Japan (1997) 9–16
9. Resnik, P.: Disambiguating Noun Groupings with Respect to WordNet senses. In Armstrong, S., Church, K., Isabelle, P., Manzi, S., Tzoukermann, E., Yarowsky, D., eds.: Natural Language Processing using Very Large Corpora. Kluwer Academic Press (1999) 77–98
10. Schaüble, P., Sheridan, P.: Cross-Language Information Retrieval (CLIR) Track Overview. In Voorhees, E., Harman, D., eds.: The Sixth Text REtrieval Conference (TREC-6), Gaithersburg, MA (1997) 31–44
11. Cutting, D., Kupiec, J., Pedersen, J., Sibun, P.: A practical part-of-speech tagger. In: Proceedings of the Third Conference on Applied Natural Language Processing, Trento, Italy (1992) 133–140
12. Robertson, S., Walker, S., Beaulieu, M.: Okapi at TREC-7: automatic ad hoc, filtering VLC and interactive track. In: NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC-7), Gaithersburg, MA (1998) 253–264
13. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley Longman Limited, Essex (1999)
14. Krovetz, R., Croft, B.: Lexical ambiguity and information retrieval. ACM Transactions on Information Systems **10** (1992) 115–141
15. Sanderson, M.: Word sense disambiguation and information retrieval. In: Proceedings of the 17th ACM SIGIR Conference, Dublin, Ireland (1994) 142–151
16. Qu, Y., Grefenstette, G., Evans, D.: Resolving translation ambiguity using monolingual corpora. In: Cross Language Evaluation Forum 2002, Rome, Italy (2002)
17. Jing, H., Tzoukermann, E.: Information retrieval based on context distance and morphology. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), Seattle, WA (1999) 90–96
18. Stevenson, M.: Augmenting Noun Taxonomies by Combining Lexical Similarity Metrics. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING-02), Taipei, Taiwan (2002) 953–959

Fault-Tolerant Fulltext Information Retrieval in Digital Multilingual Encyclopedias with Weighted Pattern Morphing

Wolfram M. Esser

Chair of Computer Science II, University of Würzburg
Am Hubland, 97074 Würzburg, Germany
esser@informatik.uni-wuerzburg.de
http://www2.informatik.uni-wuerzburg.de

Abstract. This paper introduces a new approach to add fault-tolerance to a fulltext retrieval system. The *weightedpattern morphing* technique circumvents some of the disadvantages of the widely used edit distance measure and can serve as a front end to almost any fast non fault-tolerant search engine. The technique enables approximate searches by carefully generating a set of modified patterns (morphs) from the original user pattern and by searching for promising members of this set by a non fault-tolerant search backend. Morphing is done by recursively applying so called *submorphs,* driven by a penalty weight matrix. The algorithm can handle phonetic similarities that often occur in multilingual scientific encyclopedias as well as normal typing errors such as omission or swapping of letters. We demonstrate the process of filtering out less promising morphs. We also show how results from approximate search experiments carried out on a huge encyclopedic text corpus were used to determine reasonable parameter settings.
A commercial pharmaceutic CD-ROM encyclopedia, a dermatological online encyclopedia and an online e-Learning system use an implementation of the presented approach and thus prove its “road capability”.

1 Introduction

One of the main advantages of digitally stored texts is the possibility to easily retrieve information from their content. In written texts there is always the possibility of errors and ambiguities concerning their content. Particularly large scientific encyclopedias, though they may have passed a thorough scrutiny, often use more than one spelling for the same scientific term.

Publishers of encyclopedias and dictionaries are often confronted with a problem when a large number of contributing authors produce the text content. These authors tend to use synonymous notations for the same specific term. This might seem of minor importance to the user of a printed edition. The user of an electronic version, however, might be misled by the fact that a retrieval produced results. The user might have had more results when searching for a different spelling of the search term (e.g., different hyphenation, usage of abbreviations, multilingual terms).

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 338–352, 2004.

Since 1999 our chair of computer-science has cooperated with Springer-Verlag, a well-known scientific publishing company, to compile the annual electronic version of *Hagers Handbuch der Pharmazeutischen Praxis* (Hager's Handbook of Pharmaceutic Practice) [1], the standard encyclopedia for German speaking pharmacists and pharmacologists. The printed version of "Hager's Handbook" consists of twelve volumes with about 12,300 pages. The first part (five volumes) describes active substances drawn from plants (herbal drugs), and the second (five volumes) is about synthetically produced agents (drugs). The two last volumes contain the manually generated index.

The first electronic version was released as *HagerROM 2001* at the end of 2000, and the current 3rd release was in June 2003 as *HagerROM 2003* [2]. To make the vast amount of Hager's information accessible to the end-user, a fast *q*-gram based fulltext retrieval system, which is briefly described in section 2, was built into *HagerROM.*

For a better understanding of the decisions we made during the development of the text retrieval system, some key data regarding *HagerROM* follows:

- The first edition was published by Mr. Hager in 1876
- 600 contributing authors wrote the current fifth edition (1995–2000) in
- 6,100 separate articles, which led to
- 121 MB of XML tagged text, which, in turn, lead to
- 157 MB of HTML tagged text in
- 16,584 HTML files, which resulted in
- 53 MB raw text T (after removing layout tags) with
- >160 symbol long alphabet $\boldsymbol{\Sigma}$ (after lowercasing T)

We soon were confronted with the need for making the text retrieval fault-tolerant, owing to the following characteristics of Hager: The two expert-generated index volumes of the print edition list about 111,000 different technical terms drawn manually from the Hager text corpus. These entries were used in the articles by the large number of contributing authors and consist of German, Latin and English terms – making text retrieval a multilingual problem.

For example "`Kalzium`" (Ger.) has 42 occurrences and "`Calcium`" (Lat.) has about 3750 occurrences in text T. So, whatever word variant a user feeds into a non fault-tolerant search algorithm, not all usages of the technical term will be found. Additionally spelling and typing errors are a problem in such a large text corpus. For example the word "`Kronblätter`" (crown leaves), with about 600 hits, was once typed wrong as "`Kronbläter`" and occurs once correctly as a slightly changed substring of "`kronblättrig`" (crown petaled). Empirical studies by Kukich [3] have shown that the percentage of mistakes in texts is not negligible. More precisely, she found that texts contain 1%–3.3% typing errors and 1.5%–2.5% spelling errors.

This paper is organized as follows: In section 2 we first give an overview of previous work, describe our non fault-tolerant search backend, and summarize other existing fault-tolerant approaches. In section 3 we introduce the algorithm of weighted pattern morphing and provide the rationale for the parameter choices used in the design of the algorithm. Section 4 then shows the results of some retrieval experiments we carried out on the text corpus of HagerROM, to give an idea of the average running time of the algorithm. Finally, section 5 presents conclusions and ideas of future work.

2 Previous Research

Fast search algorithms store their knowledge of a text in an appropriate index, commonly implemented using one of the following data structures (see [4]): *suffix tree, suffix array, q*grams or *q*samples (Sometimes authors refer to q-grams and q-samples as *n-grams* and *n-samples* respectively).

2.1 Our Non Fault-Tolerant Search Backend

The non fault-tolerant variant of our search engine uses a compressed *q*-gram index. When this index is created, the position (offset from first symbol in *T*) of every substring *Q* with length *q* inside text *T* is stored in an appropriate index (see [5] for details).

As the position of every substring of length *q* is stored in the index, this leads to quite large index sizes, which is seen as the main disadvantage of this indexing technique (see [4], [6]). On the other hand storage space is nowadays often a negligible factor, and so one can benefit from the enormous retrieval speed *q*-gram indices provide.

In the field of fault-tolerant information retrieval, vectors of q-grams are sometimes used to calculate the spelling distance of two given strings (see Ukkonen's q-gram distance measure [7]). But as this technique is rather oriented towards spelling and not towards sound we use weighted pattern morphing (WPM) for approximate matching. For our approach the q-gram index serves as an (exchangeable) exact, non-approximate search backend, where other data structures like suffix tree or a word index would also be feasible. In our case, q-grams are a good choice, as they are very fast in detecting that a special pattern is not part of the text: e.g., if any of the q-grams contained in the search pattern is not part of the q-gram index the algorithm can terminate – there are no occurrences of the pattern inside the text. This is useful, as many patterns that WPM generates may not be part of the text.

It is obvious that the size of the above mentioned offset lists is independent of *q,* as the position of the *Q* window always increases by one. Further, with increased values of *q,* the average length of the offset lists drops, while the total number of these lists raises, and so does the required storage space for the managing meta structure for the index.

To get reasonable retrieval performance, values of $q \geq 3$ are mandatory to avoid processing long offset lists during a search. However, when only an index for $q \geq i$ is generated, search patterns *P* with $|P| < i$ cannot be found in acceptable time. Consequently, indices for more than one *q* are needed, which leads to even more storage space requirements for the total index structure. (Note: $|X|$ denotes the length of string *X* in characters).

Instead of saving storage space by using *q*-samples, which are non-overlapping *q*-grams (i.e., every h^{th} *q*-gram, $h \geq q$, is stored in the index), we use normal, overlapping *q*-grams with *q={1,2,3,4}*. For an approximate search approach with *q*-samples see [6]. But to save space, we skip every *3*- and *4*-gram *Q* where at least one character of *Q* is not among the *f* most frequent unograms (i.e., *1*-gram) of the text, so called favorites.

So while the unogram and duogram index is complete, we skip every occurrence of, for example, 17_°, 7_°C and _°C_ (where '_' denotes 'space'), while we store every position of, for example, _rat, rats and ats_.

This technique turned out to be extremely flexible for the process of tuning our search engine to maximum speed by filling up the available storage space (e.g., of the distributed CD-ROM) with more and more *3*- and *4*-grams in our index structure.

Though the storage of unograms might seem obsolete, when duograms are present, unograms are necessary for two reasons: First, retrieval of seldom used symbols like a Greek 'δ' might be important to the end-user, as even this single symbol carries enough information content to be interesting. Second, our fault-tolerant add-on (see section 3) may modify user patterns using '?' wildcards, leaving unograms close to the borders of the new pattern.

2.2 Common Techniques for Fault-Tolerant Fulltext Retrieval

In 1918 Robert C. Russell obtained a patent for his Soundex algorithm [8], where every word of a given list was transformed in a phonetic, sound-based code. Using this Soundex code, U.S. census could look up surnames of citizens rather by sound instead of spelling, e.g., `Shmied` and `Smith` both have the Soundex code S530. Unfortunately Soundex too often gives the same code for two completely different words: Catherine and cotroneo result in C365 and similar sounding words get different codes: `night=N230` and `knight=K523`.

Although there have been many improved successors to this technique (e.g., [9] and [10]), all of them are word based and thus lack the ability to find patterns at arbitrary positions inside a text (e.g., pattern is substring of a word inside the text). Further, with sound code based systems it is impossible to rank strings that have the same code: strings are either similar (same code) or not (different code). Last, phonetic codes are usually truncated at a special word length, which make them less usable in texts with long scientific terms.

In [4] a taxonomy for approximate text searching is specified. According to this taxonomy, three major classes of approaches are known: *neighborhood generation, partitioning into exact search* and *intermediate partitioning.*

Neighborhood generationn generates all patterns $P' \in U_k(P)$ that exist in the text, where `editdistance`$(P, P') \leq k$ for a given *k* (for a description of edit distance see below). These neighbor patterns are then searched with a normal, exact search algorithm. This approach works best with suffix trees and suffix arrays but suffers from the fact that $U_k(P)$ can become quite large for long patterns *P* and greater values of *k*.

*Partitioning into exact search*carefully selects parts of the given pattern that have to appear unaltered in the text, then searches for these pattern parts with a normal, exact search algorithm and finally checks whether the surrounding parts of the text are close enough to the original pattern parts.

Intermediate partitioning, as the name implies, is located between the other two approach classes. First, parts of the pattern are extracted, and neighborhood generation is applied to these small pieces. Because these pieces are much smaller and may have fewer errors than the whole pattern, their neighborhood is also much smaller. Then exact searching is performed on the generated pattern pieces and checked to see whether the surrounding text forms a search hit.

Various approaches have been developed to combine the speed and flexibility of q-gram indices with fault-tolerance. Owing to the structure of q-gram indices, a direct neighborhood generation is not possible in reasonable time. Jokinen and Ukkonen present in [11], how an approximate search with a q-gram index structure can be realized with *partitioning into exact search.* Navarro and Baeza-Yates in [5] use the same basic approach, but assume the error to occur in the pattern, while Jokinen and Ukkonen presume the error to be in the text, which leads to different algorithms. Myers demonstrates in [12] an *intermediate partitioning* approach to the approximate search problem on a q-gram index.

All the above methods are based on the definition of one of the two string similarity metrics published by Levenshtein in [13] called *Levenshtein distance* and *edit distance.* Both metrics calculate the distance between two strings by summing up the minimal costs of transforming one string into the other by counting the atomic actions *insert, delete* and *substitute* of single symbols [14].

Though these metrics provide a mathematically well-defined measure for string similarities, they also suffer from the inability to model similarity of natural language fragments satisfactorily, from a human point of view.

With regard to the special characteristics of the Hager text corpus, the use of the edit distance measure did not seem appropriate. This is mainly due to the fact that the edit distance processes only single letters (regardless of any context information) and does not provide the means of preferring a string substitution $A \rightarrow B$ versus $A \rightarrow C$, where $|A| \geq 1$, $|B| \geq 1$ and $|C| \geq 1$ and $|A|$, $|B|$ and $|C|$ are arbitrarily different.

For example:
`editdistance`(“`kalzium`”, “`calcium`”)=2 and
`editdistance`(“`kalzium`”, “`tallium`”)=2, are the same – despite the fact that every human reader would rate the similarity of first two strings much higher than the similarity of the second pair of strings.

Because the edit distance is more suited to model random typing mistakes or transmission errors, we needed a way to approximate patterns where the differences between text and pattern are less “random” but more due to the fact that a great number of authors may use the same scientific term in different (but correct) spellings. We also wanted to cope with the problem of non-experts knowing how a scientific term sounds, without exactly knowing the correct spelling. Our technique of *weighted pattern morphing* is described in the next section.

3 Weighted Pattern Morphing

In this section we present the architecture and algorithms of our fault-tolerant frontend, which is based on the weighted pattern morphing approach. Afterwards we show the results of experiments that led to reasonable parameter settings for our fault-tolerant search engine.

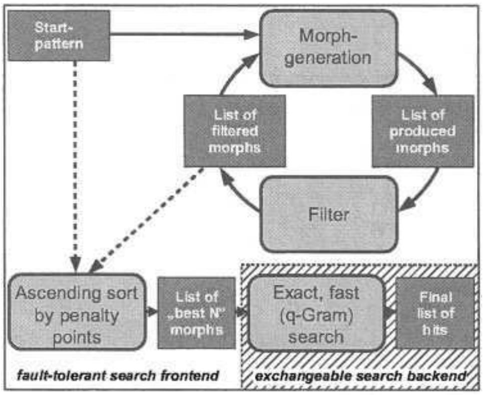


Fig. 1. Workflow of weighted pattern morphing frontend and search backend

3.1 The Fault-Tolerant Search Frontend

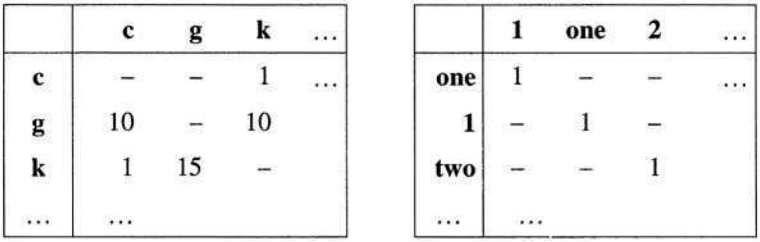
As stated at the end of the previous section the edit distance metric, which is used by most available approximate text retrieval algorithms, is not appropriate, when one is trying to model a more human-oriented string similarity. Weighted pattern morphing (WPM) overcomes the mentioned disadvantages with a simple but powerful idea:

Browse searchpattern P for all substrings $p_{i,j}$ $(1 \leq i \leq j \leq |P|)$, which are part of a phoneme group G with $G=\{g_1,g_2,\ldots,g_z\}$ and where $p_{i,j}=g_k\ (1 \leq k \leq z)$ and try to replace $p_{i,j}$ by all $g_l\ (l \neq k)$ which are members of the same phoneme group G. More general as with the edit distance, here $|p_{i,j}| \geq 1$, $|g_l| \geq 1$ and even $|p_{i,j}| \neq |g_l|$ is possible. A pattern p' where at least one substitution took place is called a *morph* of P and a single substitution of $p_{i,j}$ to g_l is called *submorph* $p_{i,j} \rightarrow g_l$.

As the interchangeability of members of the same phoneme group should be different, the concept of *penalty weights* was introduced. These penalty weights were stored in two-dimensional *submorph matrices* with source strings g_k in rows and destination strings g_l in columns (see examples in table 1).

As the table demonstrates, not every possible submorph is allowed, and the matrix may be asymmetric to the diagonal. There exist submorph tables for every common phoneme group like "a/ah/aa/ar", "i/ie/y/ih/ii", "g/j", "c/g/k/cc/ck/kk/ch", and so on. The possibilities of the edit distance can be approximated by submorphs like $\varepsilon \rightarrow$ "?" (insert any char), $c \in \Sigma \rightarrow$ "?" (substitute a char c), $c \in \Sigma \rightarrow \varepsilon$ (delete), where ε is the empty word, Σ the alphabet and "?" is the one-letter wildcard for our search engine. But even more exotic submorphs like `solution` → `sol.`, `acid` → `ac.`, `5` → `five` are defined. These are often helpful in a biochemical and medical contexts, because abbreviations are used inconsistently by different authors (e.g, in *HagerROM* the terms "`5-petaled`" and "`five-petaled`" occur).

Table 1. Two example penalty weight matrices (phoneme group "cgk..."; numbers)



	c	g	k	...
c	–	–	1	...
g	10	–	10	
k	1	15	–	
...	...			

	1	one	2	...
one	1	–	–	...
1	–	1	–	
two	–	–	1	
...	...			

For the German language mixed with Greek and Latin terms we manually identified about 25 different phoneme groups that lead to about 350 submorphs. The penalty weights for these string pairs were adjusted manually from a native speaker's point of view. Automatically adjusting the weights is subject to ongoing research, and our early results seem quite promising. For generation of English morph matrices we relied on linguistic research publications like e.g., Mark Rosenfelder's *"Hou tu pranownse Inglish"* [15]. Though Rosenfelder presents rules to get from spelling to sound, we used his work to identify about 35 English sound groups and their possible spelling which lead to English morph matrices with about 900 submorphs. Additional submorphs for number names and numbers (`100` → `hundred`, `hundred` → `100`) and domain specific abbreviations were added afterwards.

Every morphed pattern P' is recursively fed into the same morph algorithm, to perform even more submorphs. To avoid recursion loops, the first index i_{min} where submorphs $p_{i,j} \rightarrow g_l$ may start, is always increased for deeper recursion levels. Loops otherwise may appear through submorphs at different recursion levels like $u \rightarrow v$, $v \rightarrow w$, $w \rightarrow u$. On every recursion level, P is also fed unaltered into the next recursion, with only i_{min} increased, to also allow submorphs only towards the end of the pattern.

Because the recursion tree can get large, the total penalty S, as sum of the penalty weights for all applied submorphs, and M, the total number of applied submorphs (=recursion depth), are updated for every recursion call. Recursion backtracking is performed when either S or M pass configurable limits S_{max}, M_{max} or when $i_{min} > |P|$. As S_{max}, M_{max} and i_{min} grow with every recursion level, the algorithm terminates in reasonable time (see section 4).

Obviously, the above algorithm generates many morphs that are not part of the text corpus. Though the q-gram algorithm is very fast in finding out that a pattern has no hits in the text (this is so, because the search always starts with the shortest q-gram offset list, see [16]), pre-filtering of "useless morphs" was achieved by the introduction of the *hexagram filter*.

This hexagram filter possesses a trie structure with a maximum depth of six, but does not store actual offsets of hexagrams. It simply indicates whether a specific *q*-gram class (*q*≤*6*) exists in the text at all.

So when a morph *P'* is generated, the hexagram trie is traversed for every (2nd overlapping) hexagram that is part of *P'*. If any of the morph's hexagrams is not part of the trie, *P'* as a whole cannot be part of text *T* and is discarded. However, if all the hexagrams of *P'* are part of the trie, there is no guarantee that *P'* occurs in *T,* because all hexagrams are part of *T,* though not necessarily in the same order as in *P'*. In these cases we rely on the ability of the *q*-gram algorithm to terminate quickly for those patterns that are not part of the text.

When checking the q-grams of P' against the trie structure, there are two parameters that influence the accuracy of the filter: *trie depth* TD (we used a depth of six) and *window delta* WD of the hexagrams drawn from P'. The window delta states whether every hexagram of P' is taken (delta=1) or every second hexagram (delta=2) and so on. Smaller values of trie depth and larger values of window delta increase filter speed but reduce accuracy – and thus result in more promising morph candidates, which results in longer overall time for the algorithm.

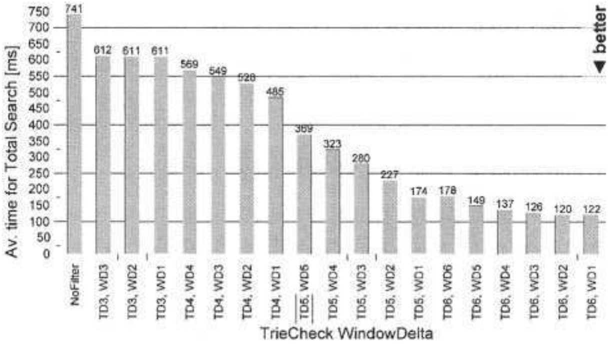


Fig. 2. Operating time for different accuracy levels of the trie filter

So, to obtain reasonable values for these two parameters, we executed fault-tolerant searches with about 14,000 patterns drawn from the text and recorded the average running times for different values of trie depth TD and window delta WD. These experiments were performed on an Intel® Pentium® IV 2.6 GHz processor with 512MB of RAM, and the results are shown in figure 2. We observed a minimum running time at TD = 6 and WD = 2, which is the reason why we chose these values for all subsequent experiments. Though these results seems portable to other Indo-European languages, it is a topic for future research whether the above values of TD and WD are appropriate for other text corpora, too.

Every time a submorph is applied, the resulting morph *P** (if it has passed the hexagram filter) is stored in a hashmap, together with S_{P*}, its sum of penalty weights. When the WPM algorithm terminates, the list of generated morphs is sorted in ascending order by S_{P*}. The best *B* morphs (those with least penalty weights) are then kept as the *final morph list* of the original pattern *P*. The limit *B* is configurable. Each triple of values S_{max}, M_{max} and *B* defines a *fault-tolerance level.*

3.2 Experiments to Determine Reasonable Parameter Settings

As stated in the previous subsection, the degree of fault-tolerance of the weighted pattern morphing algorithm can be controlled by 3 parameters:

1. $S_{max}$the maximum sum of penalty weights a morph may aggregate,
2. $M_{max}$the maximum number of submorphs a morph may accumulate, and
3. *B*the number of best rated morphs that is fed into the search backend.

The patterns an end-user presents to the search-engine remain an unknown factor, therefore we chose the following procedure to gain test patterns for our experiments: We first split up the whole text T into all of its words. As word delimiter d we chose (in perl notation):

```
d ∈ [\=\+\s\.\!\?\,\;\:\(\)\[\]\{\}\/\"\“\”\„\±\×\®\°\†\‡\…\~\'\*\·\xA0\%]
```

Words with embedded hyphens were stored as a whole and additionally all of their fragments (separated by hyphens) were added. All the words W with |W|<9 or |W|>30 were discarded. Applied to the texts of HagerROM this produced about 260,000 different words.

Every word *W* of the resulting word list WL_1 was then fed into our fault-tolerant search, while allowing very high values for S_{max}, M_{max} and *B*. All words of WL_1 where the algorithm generated morphs *P'* with $P' \in WL_1$ produced the condensed WL_2 with 14,000 different words. To minimize the runtime of the following experiments, every third word was chosen, resulting in WL_3 with about 4,600 words and an average word length of 14 chars.

So, every search pattern *P'* of WL_3 was part of the original text *T* and could be morphed (with high values for S_{max}, M_{max} and *B*), so that one or more of its morphs are again part of the total word list WL_1 – these morphs are called *valid target-morphs.* This was done to find out to what extent S_{max}, M_{max} and *B* can be decreased while keeping as many valid target-morphs as possible. The fact that only morphs *P'* with $P' \in WL_1$ were accepted in all the following experiments minimized the number of "useless" morphs. During the experiments we determined how many valid target morphs $P' \in WL_1$ the algorithm produced for a given parameter set of S_{max}, M_{max} and *B*.

The weight values for the submorph matrices were manually generated and carefully chosen from a linguistic point of view based on our experience with different multilingual text corpora (see section 5 for ideas on automatic weight generation and optimization).

Weight values were taken from integer values [*1, 2, 5, 10, 15, 20, 25, 30*] so that not every possible value was chosen, but rather "classes" of weights such as [*very-*

helpful, helpful, ..., maybe-dangerous] were used. Other numerical weight ranges are possible, but probably won't lead to better results.

The following three figures present the results of experiments where two parameters were kept fixed and the third parameter varied on each test run.

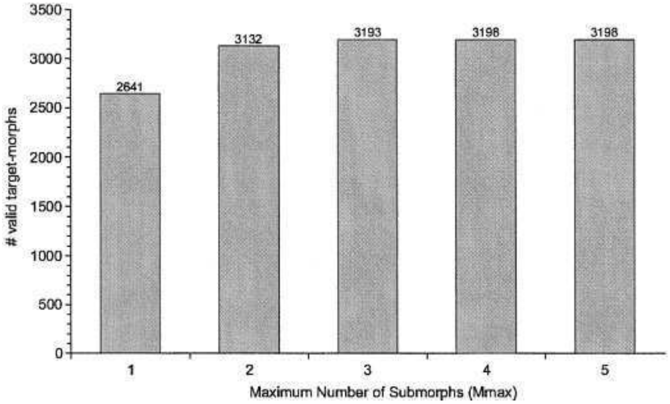


Fig. 3. Experiment#1: M_{max} variable [1,2,..., 5] (fixed: S_{max}=60, B=200)

Experiment#1 (see figure 3) led to the conclusion that M_{max} (the maximum number of applied submorphs on the original pattern) should not get greater than 4, because no increase in valid target-morphs was achieved by higher values – only more runtime was needed. The fast rise of valid target-morphs was based on the fact that S_{max} and B have quite high values in comparison to the maximum rule weight of 30.

The abrupt rise of the bar at "1 applied submorph" is due to the fact that for most word variants or words with errors only one small change (like insertion, deletion, transposition) has to be applied. Karen Kukich in [3] (see page 388) cites Mitton (1987) who examined a 170,016-word text corpus and revealed that about 70% of the misspelled words contained only a single error.

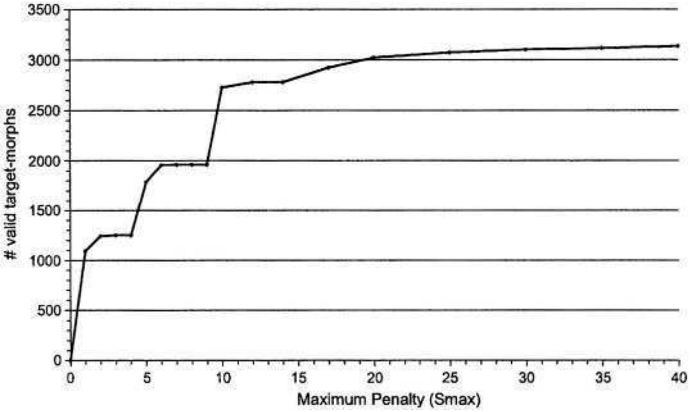


Fig. 4. Experiment#2: S_{max} variable [0, 1,..., 40] (fixed: M_{max}=2, B=200)

Experiment#2 (see figure 4) showed that Smax (the maximum sum of penalty weights a morph is allowed to collect) should not be higher than 30, which is at the same time the maximum weight used in the weight matrices. The obvious step structure of the graph in figure 4 is due to the fact that not every arbitrary weight value from the interval [1, 2,..., 29, 30] was used in the weight matrices (see above).

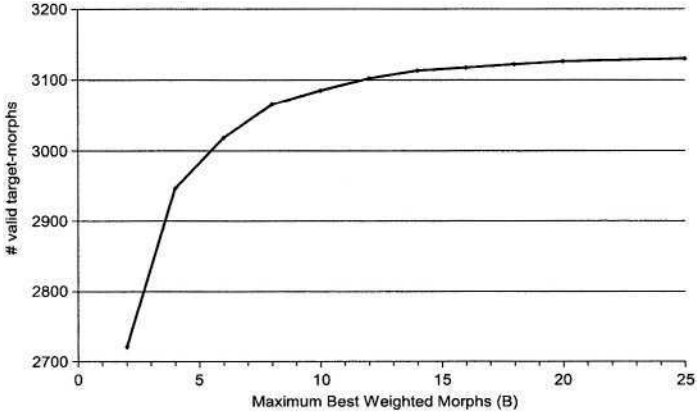


Fig. 5. Experiment#3: B variable [2, 3,..., 25] (fixed: M_{max}=2, S_{max}=60)

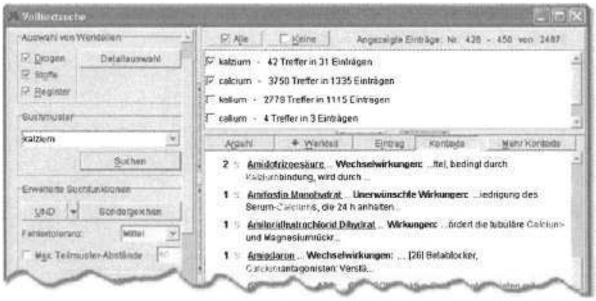
Finally, Experiment#3 (see figure 5) justifies our decision to always feed only a maximum of 20 best rated morphs to the non fault-tolerant search backend. Higher values for B may increase the overall runtime but won't improve search results any further. Note that the Y-axis of figure 5 was cut-off at a value of 2700 to allow better evaluation of the graph.

To simplify the use of the fault-tolerance feature by the end-user, macro levels labeled *low, medium* and *high* were established and grouped values for S_{max}, M_{max} and *B,* according to table 2.

Table 2. Reasonable parameter settings for different fault-tolerance levels

	low	*medium*	*high*
S_{max}	10	20	30
M_{max}	2	3	4
B	10	15	20

The graphical user interface provides the possibility to select and deselect from the list of occurring morphs, to post-filter variants of the original pattern which might be of less importance to the user. For example, a fault-tolerant search for kalzium produces also morphed hits for kalium and calium (Engl.: potassium), which is a different chemical element. The screenshot of figure 6 shows a section of the (German) user interface.



katziem

Fehlertoleranz: Mittel

Angezeigte Einträge: Nr. 426 - 450 von 2487

☑ kalzium - 42 Treffer in 31 Einträgen
☑ calcium - 3750 Treffer in 1335 Einträgen
☐ kalium - 2779 Treffer in 1115 Einträgen
☐ calium - 4 Treffer in 3 Einträgen

2 Amidotrizoesäure ... Wechselwirkungen: ...ttel, bedingt durch Kalziumbindung, wird durch ...

1 Amifostin Monohydrat ... Unerwünschte Wirkungen: ...iedrigung des Serum-Calciums, die 24 h anhalten ...

1 Amiloridhydrochlorid Dihydrat ... Wirkungen: ...fördert die tubuläre Calcium- und Magnesiumrückr...

1 Amiodaron ... Wechselwirkungen: ... [26] Betablocker, Calciumantagonisten: Verstä...

Fig. 6. *HagerROM* – Results of a Fault-Tolerant Fulltext Retrieval with WPM

4 Experiments

In this section we discuss some experiments regarding the filter efficiency and the speed of the presented fault-tolerant approach. Based on the characteristics listed in the table below, we used the text corpus of HagerROM for our experiments, because the true power of WPM shows most notably on large texts which are a real challenge to a text retrieval system. This amount of text (13 times as large as “The Bible”) and the vast number of about 600 contributing authors make the WPM based fulltext search an important part of the commercial CD-ROM product. Other examples for successful application of our WPM approach are the DEJAVU online e-Learning system and Prof. Altmeyer’s “Springer Enzyklopädie Dermatologie, Allergologie, Umweltmedizin” (Springer’s Encyclopedia on Dermatology, Allergology and Environmental Medicine). For details on DEJAVU (Dermatological Education as Joint Accomplishment of Virtual Universities), see [17]. Springer’s encyclopedia provides free online-access for physicians on [18].

Table 3. Characteristics of three products using WPM search

Module	*DEJAVU*	*Altmeyer*	*HagerROM*
Text (with Layout)	1.0 MB	22.7 MB	121 MB
Raw text (w/o Layout)	0.4 MB	5.8 MB	53 MB
Hexagram trie filter	0.3 MB	1.2 MB	6 MB
q-gram index	4.3 MB	70.2 MB	450 MB

The following table shows the results of some experiments with fault-tolerant WPM searches. The number of actual hits of a search pattern is given within parentheses. We also tested patterns that were not part of the original text, but which were transformed into valid words after passing the WPM algorithm and so, finally, produced hits in the text corpus.

Table 4. Experiments with WPM on the *HagerROM* text corpus

Original pattern	*MT sec.*	*ST sec.*	*UT sec.*	*#M*	*#F*	*#N*	*#H*	*Morphs with hits*	*# w/o filter*
azethylsalizyl **(0)**	0.23	0.12	0.53	1669	1655	14	2	acetylsalizyl(4), acetylsalicyl(435)	15035
kalzium (42)	0.05	0.01	0.23	343	336	7	5	kalzium(42), calcium(3750), kalium(2779), calium(4), cal?cium(3)	639
pneumokocken-polysacharid **(0)**	0.27	1.19	1.63	2283	2192	91	1	pneumokokken-polysaccharid (4)	129040
schokolade (54)	0.47	2.05	2.75	1578	1551	27	4	schokolade(54), shokolade(1), chocolade(1), chocolate(4)	6498
sulfamethoxy-diazin (2)	0.33	1.03	1.58	2739	2656	83	3	sulfamethoxydiazin(2), sulfametoxydiazin(17), sulfametoxidiazin(1)	24739

Legend of table 4.MT=morph time: time consumed to calculate the best #N morphs; **ST**=search time: time consumed by the non fault-tolerant search back-end to search for these best #N morphs; **UT**=user time: the total time the user has to wait for all results (with program launch time). **#M**: number of actual generated morphs; **#F**: number of morphs that did not pass the hexagram filter; **#N**: number of morphs that passed the filter with an acceptable amount of penalty weights; **#H**: number of morphs from the #N that produced at least one hit in the text corpus; **#w/o filter**: without hexagram filtering this number of (mostly useless) different morphs would have been generated.

All experiments were performed on a standard PC with AMD Athlon® 1.33GHz CPU and 512 MB RAM on a local ATA-66 harddisk under Windows XP®. The compressed q-gram index q={ 1,2,3,4} needs about 450MB storagespace (this is 8 times |T|) and can be generated on an ordinary Linux computeserver in about one hour.

Table 4 demonstrates that on an average PC hardware, fault-tolerant text retrieval with practical search patterns can be accomplished using the approach of weighted pattern morphing in acceptable time. Within the presented examples the user has to wait an average of two seconds to obtain the wanted results. The hexagram trie filter prevents the algorithm from generating thousands of morphs that can't be part of the text and thus contributes to a faster response of the system.

From our discussion it is obvious that the filter becomes less accurate with longer search patterns. This is due to the fact that the filter can only determine that every six character substring of a morph is part of text *T*. The filter can't determine whether these existing six character substrings of the morphed pattern also occur in the same order and at the same distances inside text *T*.

5 Conclusion and Future Work

We demonstrated that nowadays average end-user PCs are capable of performing multiple, iterated, exact text retrievals over a set of morphed patterns and thus simulate a fault-tolerant search. Morph matrices with penalty weights seem much more suitable and flexible to model phonetic similarities and spelling variants in multilingual, multi-author texts than the edit distance metric or phonetic codes like Soundex and its successors. Weighted pattern morphing can generate edit distance like spelling variants (delete or swap letters, insert "?" one-letter wildcards) and the algorithm can also put emphasis on phonetic aspects like sound-code based algorithms. It thus combines the strength of these two approaches.

The presented algorithm can be added on top of any exact search engine to create a fault-tolerant behavior. A q-gram index fits extremely well as exact non-fuzzy search backend, because a "no-hit" result can be detected in short time and wildcards ("?", "*") are easy to implement without extra time costs.

It will be part of future research to automatically fine-tune the penalty weights in order to customize the system to a special text. We are planning to run large test series and keep track of how often a submorph produced a valid target-morph. The collected data will enable us to fine-tune submorph weights for even better performance.

References

1. Bruchhausen F.v. et al. (eds.): *Hagers Handbuch der Pharmazeutischen Praxis. 10(+2) Bände. u. Folgebände.* Springer Verlag, Heidelberg (1992-2000)
2. Blaschek W., Ebel S., Hackenthal E., Holzgrabe U., Keller K.,Reichling J. (eds.): *HagerROM 2003 - Hagers Handbuch der Drogen und Arzneistoffe. CD-ROM.* Springer Verlag, Heidelberg (2003) http://www.hagerrom.de
3. Kukich K.: *Technique for automatically correcting words in text.* ACM Computing Surveys 24(4) (1992) 377-439
4. Navarro G., Baeza-Yates R., Sutinen E., Tarhio J.: *Indexing Methods for Approximate String Matching.* IEEE Bulletin of the Technical Committee on Data Engineering, Vol. 24, No. 4 (2001) 19-27
5. Navarro G., Baeza-Yates R.: *A Practical q-Gram Indexfor Text Retrieval Allowing Errors.* CLEI Electronic Journal, Vol. 1, No. 2 (1998) 1
6. Sutinen E.: *Filtration with q-Samples in Approximate String Matching.* LNCS 1075, Springer Verlag (1996) 50-63
7. Ukkonen, E.: *Approximate string-matching with q-grams and maximal matches.* Theoretical Computer Science 92 (1992) 191-211
8. Russell R.: *INDEX (Soundex Patent).* U.S. Patent No. 1,261,167 (1918) 1-4
9. Zobel J., Dart Ph.: *Phonetic String Matching: Lessons from Information Retrieval.* ACM Press: SIGIR96 (1996) 166-172
10. Hodge V., Austin J.: *An Evaluation of Phonetic Spell Checkers.* Dept. of CS, University of York, U.K. (2001)
11. Jokinen P., Ukkonen E.: *Two algorithms for approximate string matching in static texts.* LNCS 520, Springer Verlag (1991) 240-248
12. Myers E.: *A sublinear algorithm for approximate keyword searching.* Algorithmica, 12(4/5) (1994) 345–374

13. Levenshtein V.: *Binary codes capable of correcting deletions, insertions, and reversals.* Problems in Information Transmission 1 (1965) 8-17
14. Stephen G.: *String Searching Algorithms, Lecture Notes Series on Computing, Vol. 3.* World Scientific Publishing (1994)
15. Rosenfelder M.: *Hou tu pranownse Inglish* http://www.zompist.com/spell.html (2003)
16. Grimm M.: *Random Access und Caching für q-Gramm-Suchverfahren.* Lehrstuhl für Informatik II, Universität Würzburg (2001)
17. Projekt DEJAVU: *Homepage* http://www.projekt-dejavu.de (2003)
18. Altmeyer P., Bacharach-Buhles M.: *Springer Enzyklopädie Dermatologie, Allergologie, Umweltmedizin.* Springer-Verlag Berlin Heidelberg (2002) http://www.galderma.de/anmeldung_enz.html

Measuring a Cross Language Image Retrieval System

Mark Sanderson, Paul Clough, Catherine Paterson, and Wai Tung Lo

Department of Information Studies, University of Sheffield, S1 4DP, UK
m.sanderson@shef.ac.uk

Abstract. Cross language information retrieval is a field of study that has received significant research attention, resulting in systems that despite the errors of automatic translation (from query to document), on average, produce relatively good retrieval results. Traditionally, most work has focussed on retrieval from sets of newspaper articles; however, other forms of collection are being searched: one example being the cross language retrieval of images by text caption. Limited past work has established, through test collection evaluation, that as with traditional CLIR, image CLIR is effective. This paper presents two studies that start to establish the usability of such a system: first, a test collection-based examination, which avoids traditional measures of effectiveness, is described and results from it are discussed; second, a preliminary usability study of a working cross language image retrieval system is presented. Together the examinations show that, in general, searching for images captioned in a language unknown to a searcher is usable.

1 Introduction

A great deal of research is currently conducted in the field of Cross Language Information Retrieval, where documents written in one language (referred to as the target language) are retrieved by a query written in another (the source language). Until now, most work has concentrated on locating or creating the resources and methods that automatically transform a user's query into the language of the documents. With the right approach, CLIR systems are relatively accurate: managing to achieve retrieval effectiveness that is only marginally degraded from the effectiveness achieved had the query been manually translated (referred to as monolingual retrieval). For example, Ballesteros (1998) achieved CLIR effectiveness at 90% of monolingual retrieval.

One area of CLIR research that has received almost no attention is retrieving from collections where text is used only to describe the collection objects, and the object's relevance to a query are hopefully clear to anyone regardless of their foreign language skills. One such collection is a picture archive where each image is described by a text caption. Retrieval from such an archive presents a number of challenges and opportunities. The challenges come from matching queries to the typically short descriptions associated with each image. The opportunities derive from the unusual situation of potentially building a CLIR system that a large number people may wish to use. For any vendor of an image library, use of CLIR offers the opportunity of expanding the range of potential searchers of (even purchasers from) their library.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 353–363, 2004.

In the rest of this document, previous work in cross language and in image retrieval is described, along with past image CLIR research. This is followed by the design and results of the first test collection based study, which led onto a preliminary targeted usability experiment, which is also outlined. Finally, overall conclusions and avenues for future work are provided.

2 Previous Work

Because there is little past work investigating cross language image retrieval, this Section will first review the two component research areas separately: image retrieval by associated text and cross language IR; followed by an examination of the proposals and occasional attempts at performing image CLIR.

2.1 Image Retrieval

Retrieval of images by text queries matched against associated text has long been researched. As part of his PhD investigating multimedia retrieval, Dunlop examined such an approach to image retrieval (1993). The ideas in this work were later extended to a study of image retrieval from art gallery Web sites by Harmandas et al (1997), who showed that associated text was well suited for retrieval over a range of query types. At the same conference, research was presented on the successful use of a thesaurus to expand the text of image captions (Aslandogan et al 1997). More recently, research in combining content-based image retrieval with caption text has been explored in the work of Chen et al (1999).

2.2 Cross Language Retrieval

Most CLIR research effort has focussed on locating and exploiting translation resources. Successful methods centre on use of bilingual dictionaries, machine translation systems, or so-called parallel corpora, where a large set of documents written in one language are translated into another and word translations are derived from their texts. With reasonably accurate translation, effective cross language retrieval is possible. This is being confirmed in the large-scale evaluation exercises within TREC[1] and CLEF[2]. A thorough review of this aspect of CLIR research can be found in Gollins (2000).

2.3 Cross Language Image Retrieval

The potential utility of image CLIR has been known about for sometime: both Oard (1997) and Jones (2001) discussed the topic. However, only a few examples of image CLIR research exist.

[1] http://trec.nist.gov/

[2] http://www.clef-campaign.org/

1. The IR Game system built at Tampere University (Sormunen, 1998) offered Finish/English cross language retrieval from an image archive, images were ranked using a best match search. However, little has been written about this system.
2. The European Visual Archive (EVA[3]) offered English/Dutch/German cross language searching of 17,000 historical photographs indexed by a standard set of 6,000 controlled terms. Searching is restricted to Boolean searching.
3. Flank (2002) reported a small scale test collection style study of image CLIR using a collection of 400,000 images captioned in English and ten queries translated into three languages. She concluded from her results that image CLIR was effective. However, the conclusions were based on a collection that by most standards is too small: Voorhees (1998) emphasised the importance of working with test collections that have at least 25 queries; any fewer and results derived maybe unreliable.
4. In 2002, the imageCLEF track of the CLEF (Cross Language Evaluation Forum) exercise was established with the release of one of the first publicly available image test collections: consisting of approximately 30,000 images (from a digital collection held at St. Andrews University in Scotland) and fifty queries. The collection was used by four research groups. Working across a number of European languages – and in the case of one research group, Chinese – it was shown that cross language retrieval was operating at somewhere between 50% and 78% of monolingual retrieval (Clough, 2003), confirming Flank's conclusion that image CLIR can be made to work.

From these past works, one might conclude that image CLIR is feasible; however, the conclusion would be based on test collection evaluation alone. It is also necessary to consider if user will be able to judge retrieved images as relevant.

Judging Images for Relevance

In past works where image CLIR has been suggested as an application of cross language technologies (Oard, 1997 and Jones, 2001), there has been an assumption that users are generally capable of judging the relevance of images simply by observing them: reading caption text is not necessary. It is to be anticipated, however that there will be limits to such a capability. For example, in the two images shown below, most would agree the first was relevant to a query for mountain scenery without having to read any caption; however, for a query requiring a picture of London Bridge, the only way most searchers would know if the second image was relevant or not was by reading an associated caption. This problem was highlighted in an extensive study of image retrieval by Choi and Rasmussen (2002) who showed user's judgement of image relevance was often altered once they read the image's caption (p. 704). Aware of this issue, Flank (2002) in her small test collection-based experiment, limited the scope of the ten queries used to a generic set of topics that she believed most users would be able to judge image relevance from.

It appears that in order for an image CLIR system to work well, it will be necessary to include a translation of captions in order to provide users as much information as possible to judge relevance. In text document CLIR, Resnik reported his early work on so-called gists of documents (1997), however evaluation of the system was such that it is hard to speculate on users' abilities to judge relevance from the gists. This

[3] http://www.eva-eu.org/

ability was more thoroughly assessed by Gonzlao et al (2004) who reported the results of the CLEF interactive track. Here it was shown that users were capable of determining the relevance of retrieved documents accurately after they were automatically translated in some manner.

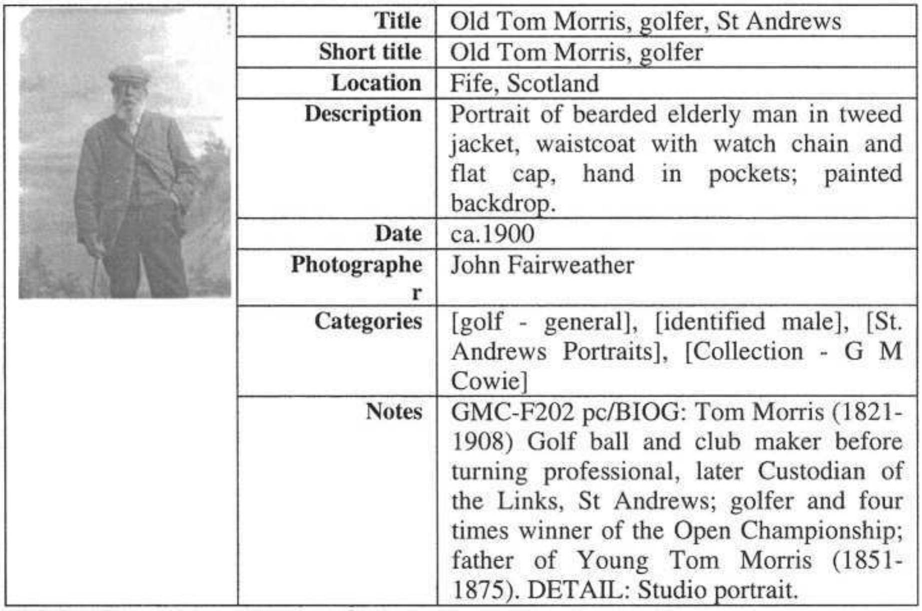
3 Problems with Test Collection Measures

Effectiveness of retrieval system is almost always measured on a test collection using either average precision measured across a series of recall values or at a fixed rank. It would appear that this is done as these measures are commonly assumed to be a reasonable model of a user's view of the effectiveness of a retrieval system: the higher the average precision, the more satisfied users will be. How true this view is appears, perhaps surprisingly, not to have been tested as thoroughly as one might expect. For example, techniques exist, such as pseudo relevance feedback, that consistently produce higher average precision than baseline systems, implying that they would be preferred by users, but they are rarely deployed in actual IR systems. Such a contradiction suggests that, in this case at least, average precision is not reflecting user preferences. In CLIR, effectiveness is measured as the percentage reduction in average precision from that achieved with monolingual queries. With the result from Clough & Sanderson (a lower bound of 50% of monolingual), one may conclude that for every ten relevant documents resulting from a monolingual search, five are retrieved for every query tested. However, the consistency of the reduction in effectiveness is uneven, as a study of individual queries reveals.

In order to better understand the unevenness for image CLIR, a test collection evaluation was conducted where the measures of average precision and precision at rank ten were dropped in favour of a measure perceived to be more "user centred". Before describing the experiment, the collection that searching was conducted on is first outlined.

3.1 The Collection

Selecting a suitable collection for image retrieval was a non-trivial task. Not only was a "large" collection of images required, but also a collection with captions of high quality to facilitate text-based retrieval methods. Additional issues involving copyright were also encountered as typically photographs and images have a potentially high marketable value, thereby restricting permissible distribution. A library that was willing to release its collection was eventually found. St. Andrews University[4] holds one of the largest and most important collections of historic photographs in Scotland, exceeding over 300,000 photographs from a number of well-known Scottish photographers (Reid, 1999). A cross-section of approximately 30,000 images from the main collection was digitised to enable public access to the collection via a web interface. Permission was sought and granted by St. Andrews Library to downloaded and distribute the collection for research use.



Title	Old Tom Morris, golfer, St Andrews
Short title	Old Tom Morris, golfer
Location	Fife, Scotland
Description	Portrait of bearded elderly man in tweed jacket, waistcoat with watch chain and flat cap, hand in pockets; painted backdrop.
Date	ca.1900
Photographer	John Fairweather
Categories	[golf - general], [identified male], [St. Andrews Portraits], [Collection - G M Cowie]
Notes	GMC-F202 pc/BIOG: Tom Morris (1821-1908) Golf ball and club maker before turning professional, later Custodian of the Links, St Andrews; golfer and four times winner of the Open Championship; father of Young Tom Morris (1851-1875). DETAIL: Studio portrait.

This collection was used as the basis for ImageCLEF and the experiments described here. The collection consists of 28,133 images (a 368 by 234 pixel large version along with a 120 by 76 thumbnail[5]) and captions. The majority (82%) of images are in black and white ranging between the years 1832 and 1992 (with a mean year of 1920). Images and captions of varying styles, presentation, and quality exist in the collection. The figure below shows an example image and caption from the collection. The captions consist of data in a semi-structured format added manually by domain experts at St. Andrews University. The caption contains eight fields, the most important being the description, which is a grammatical sentence of around fifteen words. The captions exist only in British English, and the language tends to contain colloquial expressions.

[4] http://www-library.st-andrews.ac.uk/

[5] Much higher resolution images are retained by St. Andrews and are not part of the collection.

3.2 Test Collection Study

As part of preparations for the formation of the imageCLEF collection (Clough and Sanderson, 2003), a preliminary evaluation of image CLIR was conducted on the St. Andrews collection of images, the full details of which are described by Paterson (2002). A set of fifty queries was created in English, which were manually translated into Portuguese and German. The queries were then translated back into English using AltaVista's Babel Fish[6] and submitted to an in-house searching system which uses a BM25 weighting scheme, searching the full text captions of the collection. With similar results to the imageCLEF experiments described above, the conclusions of the study revealed that CLIR of German and Portuguese retrieved respectively between 55% and 67% of the relevant documents retrieved by monolingual searching. As with imageCLEF, the results here were measured with a form of average precision, in this case, precision at rank ten. Average precision measures, across queries, the density of relevant documents measured at certain fixed positions: either at a fixed rank position, as used here; or measured at standard levels of recall as is commonly found in recall/precision graphs. As already described in this paper, it is not clear how well average precision models user preferences. Consequently, an alternative measure was sought. In an attempt to derive one, a consideration of what was important to searchers when using retrieval systems was conducted. Three cases were considered and user reactions conceived:

1. Returning one relevant document: at a minimum, users will be content with finding a single relevant item returned by their query. If the cross language system offers relevance feedback, users are likely to be able to locate more relevant items through use of that process;
2. Returning more than one relevant document: although additional user satisfaction would inevitably result from a larger number of relevant items being retrieved, the benefits to users of this aspect is not as important as the detriments of the following situation;
3. Returning no relevant documents: a great deal of user dissatisfaction would be associated with no relevant documents being retrieved. This may be particularly important for cross language retrieval, as reformulation of a query is likely to be harder than with monolingual searching.

The conclusion of this consideration was that emphasising user dissatisfaction at failed queries in the effectiveness measure is the priority. Consequently, it was decided that simply counting the number of queries resulting in at least one relevant image being returned in the top ten would be an effective measure of user satisfaction as it would better reflect user preferences over the range of searching situations as well as being a clear and simple measure to understand. The following table shows the effectiveness of the retrieval system across the three languages as measured by successful queries and, for comparison, by average precision at rank ten.

[6] http://babelfish.altavista.com/, service used in the summer of 2002.

1	2	3	4	5	6
Query language	**Query retrieves ≥1 relevant**	**Absolute %**	**Relative %**	**Precision at rank 10, %**	**Relative %**
English	48	96	100	49	100
German	35	70	73	27	55
Portuguese	39	78	81	33	67

It is our contention that in comparison to averaged precision (column 5), the measure of successful queries (column 2) provides a clearer notion of the user experience when searching: to see that two in fifty queries fail with English in comparison to the fifteen in fifty for German is more tangible (we would claim) than observing the density of relevant documents in the top 10 reduced from 49% to 27%. It is also worth noting that the reduction in effectiveness, measured by successful queries (column 4) showed a smaller reduction in effectiveness from monolingual than that measured with precision (column 6). From the data in the table, we draw two conclusions

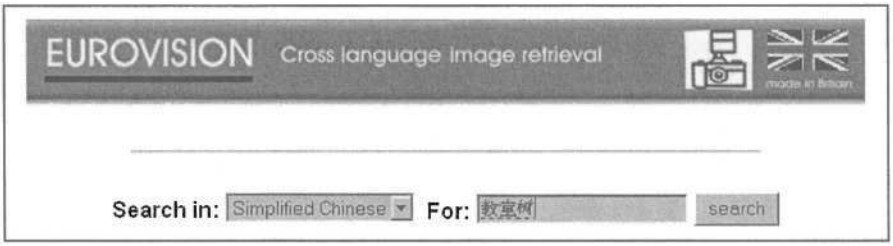
1. When considering the number of failed queries in cross language retrieval, it can be seen that a significant number of queries fail.
2. Although the failure rate is high, it is notable that although the density of relevant images retrieved drops considerably, from monolingual to cross language the number of "failed queries" does not drop as much.

The overall conclusion of this experiment is that as with previous research, image cross language retrieval is workable, however, by analysing the number of queries that return no relevant images in the top 10, it is clear that users of such a system will be faced with the problem of having to deal with failed queries on a reasonably frequent basis.

To further understanding of image CLIR, however, it was decided a usability test be conducted to examine how users may cope with querying for images across languages. Before describing the test, however, the image CLIR system that the test was conducted on is first described.

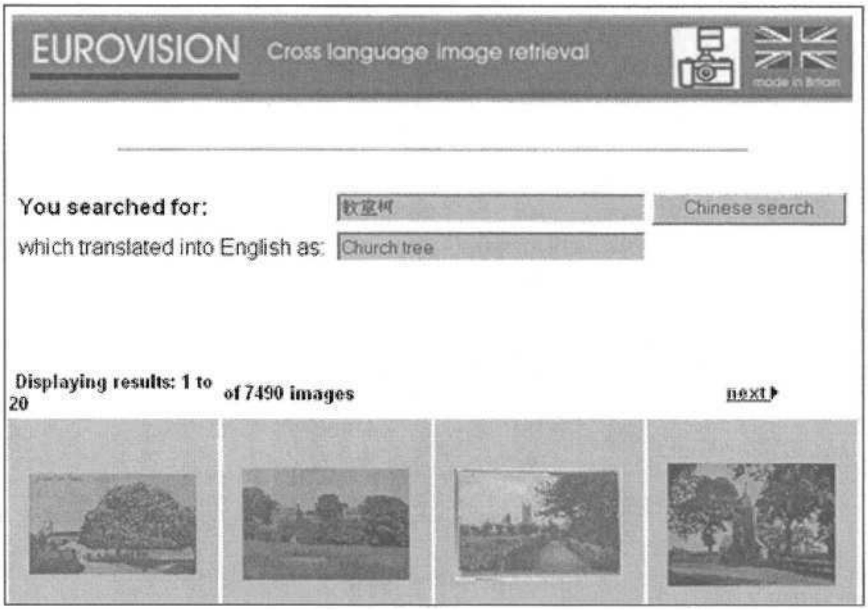
4 The Image CLIR System

The system tested was an image caption-based text retrieval system created "in-house" by the authors. It offers a straightforward search engine-like interface, where users enter their search terms in their native language (in the figure below, Chinese).



Search in: Simplified Chinese For: 教堂树

As with the earlier test collection study described above, the AltaVista Babel Fish translation system was accessed when translating a user's query from the source to the target language (i.e. Chinese to English). Retrieval on the source language version of the query was performed by an IR system searching on all parts of the image captions, using BM25 to rank images. As can be seen below, both the original query and its translation (correct in this case) are displayed along with the retrieved images. The retrieved set is shown as a table of thumbnails grouped in batches of twenty. To avoid the risk of over using the Babel Fish service, translation of image captions was not performed.



You searched for: 教堂树

which translated into English as: Church tree

Displaying results: 1 to 20 of 7490 images

Users were free to re-enter or modify queries as they preferred.

4.1 The Experiment

The usability experiment was set up as a preliminary exploration of how capable users were at cross language image searching. It was decided to set a series of known item searching tasks. Subjects were shown an image (without any caption text) and asked to find that image within the collection. This caused them to try to find a form of query words that would help them locate the image. The fact that captions of retrieved images in the users' language were not available was less of a problem as judgement of relevance of the known item could be made without recourse to caption text. It was expected that searchers would be willing to put more effort into searching and reformulation than a more classic IR "find images relevant to topic x" form of task as they knew the image was within the collection and knew they hadn't succeeded until they found it.

The Subjects

As the collection being searched was captioned in English, the queries were written in (and the searchers had to be fluent in) a different language. With such a collection, it would be preferable for the searchers not to know English. Locating such people within the UK proved to be too hard a task. However, it was possible to locate a large number of bilingual people. For the usability experiment, eighteen native Chinese language speakers were recruited. They were students at the University of Sheffield taking a postgraduate Masters course; each was paid £15 for participating. As with any usability test, the time one can take with each subject was limited by the amount of time someone can reasonably be expected to continually search before tiring. Therefore, each searcher spent approximately one hour conducting the experiment, completing pre and post test questionnaires and being interviewed.

The pre-test questionnaire established that the subjects felt they were good at searching; most of them making at least weekly access to either a search engine or electronic library catalogue. A problem, for the experimental setup with these subjects was that by their own admission their command of English was good (66%) or fair (22%); only two (12%) regarding their English as basic. With such language skills, if the subjects viewed the English captions of the retrieved images, they would become frustrated by having to search in Chinese through a translation system, perhaps preferring to re-formulate their queries in English. Therefore, it was decided that the captions of the retrieved images would be removed from display of the interface. As the task chosen for users was a known item task, captions were not needed to help with judging relevance, therefore, their removal for this type of experiment was judged not to be problematic.

Experiment

A series of images were selected to give users a range of difficulty of locating a known item. The images were of a bridge, a ship, a Tapir, an empty street, and a person:

- A bridge, and a ship – these images were expected to be relatively easy to query for;
- the Tapir, less so, as users may not know the name of this animal;
- the person and street scene would be hard to locate without knowing the name of the street or the town it was in.

No time limit was placed on the users' execution of each known item search. The number of queries issued, the number of search result pages viewed (in total), the time taken, and the success rate of the experimental subjects was logged.

Results

As expected, across the five known items, success varied: for three of the five images nearly all users were able to locate the known item. The image of the bridge was hard to locate as the collection holds a pictures of a great many. Users consequently searched through a large number of result pages (eighteen on average). Determining an appropriate query to retrieve the street scene image proved almost impossible for users, although a number of queries (on average six) were tried by the experimental subjects in their attempts to locate it; only one user succeeded. Most users located the image of the person, though as with the bridge, many images of people are to be found in the St. Andrews collection.

Image	*Bridge*	*Ship*	*Tapir*	*Street*	*Person*	Average
Queries	3.4	2.8	1.8	6	4.3	3.66
Pages viewed	18	2.6	1	22	18.7	12.46
Time (min.)	6.2	1.7	1.4	7.0	7.8	4.82
Success (%)	88.9	100	100	5.6	66.7	72.24

A number of conclusions about image CLIR were drawn from the results of this preliminary test:

- Users appear to be relatively successful in known item searching within an image CLIR system. The tasks set were not trivial (especially in the case of the final two images) and one would not expect users to be 100% successful even if they were searching in English. It is to be hoped that the success observed here will transfer to more general image CLIR tasks.
- Users appear to be willing to re-submit and re-formulate queries to a CLIR system in order to locate an item.
- Users are willing to look through a large number of retrieved images in order to find what they are seeking.

The overall conclusion of the work was that users are capable of searching for images in a CLIR system with a relatively high degree of success.

5 Conclusions and Future Work

Two experiments were described where the feasibility of image CLIR was examined. First, a test collection-based study explored a different means of measuring the effectiveness of a CLIR system. It was argued that the measure better illustrated the problems with CLIR, namely queries that fail to retrieve any relevant images. Second, a small usability study of a working image searching system was tested with users querying in a language different from that of the image captions. Here, it was concluded that users were more than capable of searching for items in a collection; a conclusion that bodes well for CLIR when applied to image collections.

For future work, the effectiveness of automatic translation of image captions will be examined and consequently a wider ranging usability test will be conducted to broaden the set of tasks users are observed completing. A re-examination of existing CLIR research is planned where measuring and comparing past results using the failed queries statistic will be conducted.

Acknowledgements. This research was supported in part by the EPSRC fast track grant Eurovision (GR/R56778/01).

References

Aslandogan, Y.A., Thier, C., Yu, C.T., Zou, J., Rishe, N. (1997) Using Semantic Contents and WordNet in Image Retrieval in Proceedings of ACM SIGIR '97, 286-295.

Ballesteros, L., Croft, W.B. (1998): Resolving ambiguity for cross-language retrieval, in Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, J. Zobel (eds.): 64-71

Chen, F., Gargi, U., Niles, L., Schuetze, H. Multi-Modal Browsing of Images in Web Documents, Proceedings of SPIE Document Recognition and Retrieval VI, pp. 122-133, 1999

Choi, Y., Rasmussen, E.M. (2002) Users' relevance criteria in image retrieval in American history. Information Processing and Management 38(5): 695-726

Clough, P., Sanderson, M. (2003) The CLEF 2003 Cross Language Image Retrieval Task, in Working Notes for the CLEF 2003 Workshop, 21-22 August, Trondheim, Norway

Dunlop, M.D. & van Rijsbergen, C. J. (1993) Hypermedia and free text retrieval, Information Processing and Management, vol 29(3).

Flank, S. (2002) Cross-Language Multimedia Information Retrieval, in the Proc. 6th Applied Natural Language Processing Conference

Gollins, T. (2000) Dictionary Based Transitive Cross-Language Information Retrieval Using Lexical Triangulation, Masters Dissertation, Department of Information Studies, University of Sheffield, Regent Court, 211 Portobello Street, S1 4DP, Sheffield.

Harmandas, V, Sanderson, M., Dunlop, M.D. (1997) Image retrieval by hypertext links In the Proceedings of the 20th ACM SIGIR conference, Pages 296-303, 1997

Jones, G.J.F., New Challenges for Cross-Language Information Retrieval: Multimedia Data and the User experience. In Carol Peters (ed.). Cross-Language Information Retrieval and Evaluation: Proceedings of the CLEF 2000 Workshop, Lecture Notes in Computer Science 2069, Springer 2001, pp 71-81.

Oard, D. (1997) Serving Users in Many Languages: Cross-Language Information Retrieval for Digital Libraries In D-Lib Magazine, http://www.dlib.org/

Oard, D., Gonzalo, J., Sanderson, M., López-Ostenero, F., Wang, J. (2004) Interactive Cross-Language Document Selection to appear in the journal of Information Retrieval

Paterson, C. (2002) The effectiveness of using machine translators to translate German and Portuguese into English when searching for images with English captions, Masters Dissertation, Department of Information Studies, University of Sheffield, Regent Court, 211 Portobello Street, S1 4DP, Sheffield.

Reid, N. (1999) The photographic collections in St. Andrews University Library. Scottish Archives, Vol. 5, 83-90

Resnik, P. (1997) Evaluating Multilingual Gisting of Web Pages, in AAAI Spring Symposium on Cross-Language Text and Speech Retrieval Electronic Working Notes

Sormunen, E., Laaksonen, J., Keskustalo, H., Kekäläinen, J., Kemppainen, H., Laitinen, H., Pirkola, A., Järvelin, K., (1998) The IR Game - A Tool for Rapid Query Analysis in Cross-Language IR Experiments. PRICAI '98 Workshop on Cross Language Issues in Artificial Intelligence. pp. 22-32.

Voorhees, E. (1998): Variations in Relevance Judgements and the Measurement of Retrieval Effectiveness, in Proceedings of the 21st annual international ACM-SIGIR conference on Research and development in information retrieval: 315-323

An Optimistic Model for Searching Web Directories

Fidel Cacheda[1] and Ricardo Baeza-Yates[2]

[1] Department of Information and Communication Technologies, University of A Coruña
Facultad de Informática, Campus de Elviña s/n, 15071 A Coruña, Spain
fidel@udc.es

[2] Center for Web Research, Department of Computer Science, University of Chile
Blanco Encalada 2120, Santiago, Chile
rbaeza@dcc.chile.cl

Abstract. Web directories are taxonomies for the classification of Web documents using a directed acyclic graph of categories. This paper introduces an optimistic model for Web directories that improves the performance of restricted searches. This model considers the directed acyclic graph of categories as a tree with some "exceptions". The validity of this optimistic model has been analysed by developing and comparing it with a basic model and a hybrid model with partial information. The proposed model is able to improve in 50% the response time of a basic model, and with respect to the hybrid model, both systems provide similar response time, except for large answers. In this case, the optimistic model outperforms the hybrid model in approximately 61%. Moreover, in a saturated workload environment the optimistic model proved to perform better than the basic and hybrid models for all type of queries.

1 Introduction

The World Wide Web is a fast growing wide area hypermedia database that can be seen as a giant library containing documents from all over the world. Like any library, it must have a good cataloguing system in order to retrieve information from it. In fact, the first information retrieval (IR) system on the Web appeared and quickly became essential as the published information increased. The estimated number of pages in 1999 was 800 million, generating 6 terabytes of text information [1] and nowadays Google [9] claims to have indexed more than three billion pages.

Information retrieval systems appear in the Web with the purpose of managing, retrieving and filtering the information available in the Web. In a general way, we could distinguish three different types of search tools to locate information in the Web: search engines, Web directories and metasearch systems [2]. Search engines index one part of the documents available in the Web (ideally, the whole Web), placing quantity above quality of contents (e.g. Google [9]). Web directories classify the most relevant Web documents according to topic, creating an ontology of the Web and placing quality above quantity of documents (e.g. Yahoo! [16] or the Open Directory Project [13]). Finally, a metasearch system simply resends queries to other search systems, and later they merge and reorder the results.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 364–377, 2004.

Search engines and metasearch systems are simply based on the search process in order to retrieve information. On the other hand, Web directories allow information retrieval by means of standard searches, browsing categories and a combination of both. The browsing process consists of navigating through the category structure examining the available documents. This category structure is defined as a directed acyclic graph (DAG), since one category may have one or more children and one or several parents and, one document may be catalogued in several categories. This provides Web directories with a great power and cataloguing flexibility.

However, Web directories have the added value of a search process combined with navigation, which improves the quality of the results. In this case, the search is restricted to those documents linked to an area of the DAG specified by the user. We call that type of searches on a subset of the graph, a *restricted search.*

The search process is based on an inverted file structure that relates keywords to their associated documents, while the navigation process is based on an inverted file that associates each category to its documents. The search process restricted to one area of the DAG of categories must combine the results of both lists in an efficient way.

In this paper we analyse the characteristics of the restricted searches in monolingual Web directories, with special emphasis on the data structures used and the performance obtained. Also the hybrid model, described in [6], is analysed and a new model named optimistic model is proposed. This optimistic model will improve the response time over a basic model in approximately 50% and performs better than the hybrid model for large answers. Also, the optimistic model will obtain the best results in a saturated workload environment, a typical case in the Web.

The paper is structured as follows. First the state of the art is exposed. Section three describes the hybrid model. Next, the optimistic model is described and we give the details of the implementation and evaluate the performance of the proposed model. Finally, conclusion and areas for further research are given.

2 State of the Art

A Web directory consists of four basic components that represent the information stored in it: keywords, documents and categories; and their relationships (see Figure 1) [4, 11, 12].

For the categories, a single identifier is assigned to each of them. Thus, the basic information (name, brief description, etc.) shall be stored in a category file. Independently, a pointer-based structure based on the category identifiers shall represent the DAG that constitutes the ontology.

With regard to the documents, their basic information is stored in a document file (URL, title and content description), the order of which is based on the identifiers to facilitate access.

The set of keywords derived from the index process of both, documents and categories, is stored in the vocabulary using inverted indexes [7, 10], as the indexing technique that has the best performance for huge data volume [18].

On the other hand, their relationships must also be stored. In the case of the keywords and documents relationship, the inverted lists may present different organizations that directly influence the performance of the standard search process, and, in consequence, of the restricted searches. A classification according to the document identifier facilitates the combination of several lists, whilst a classification according to the criterion of document relevance makes simple searches trivial. There are also mixed alternatives, such as the one defined at [4], used in Google.

In Web directories also the categories are indexed, and so an inverted file is linked to the vocabulary, but stored independently from the documents inverted lists for reasons of efficiency and system complexity.

A structure that relates each category to the associated documents is also necessary, which is intensely used during the navigation process. In this case, the inverted file structure also constitutes the most efficient option, so that the category file constitutes the vocabulary, while an inverted list of document identifiers is associated to each category. This list is preferably sorted according to the relevance or importance of each document to optimise the browsing process.

The use of these structures for the standard search process is undoubted, being widely used in different traditional commercial systems and in the Web [2]. Basically, the search process is divided into three steps. First, each term of the search chain is searched in the vocabulary. Second, occurrences associated to each term must be retrieved and, finally, the different lists obtained must be combined, taking into account Boolean or proximity operations.

The navigation process is much simpler than the previous one. In this case, only one category is located (based on the identifier) and its list of associated documents is accessed, without any kind of merging operation (with an optimal performance if the list has been previously sorted according to a relevance criterion).

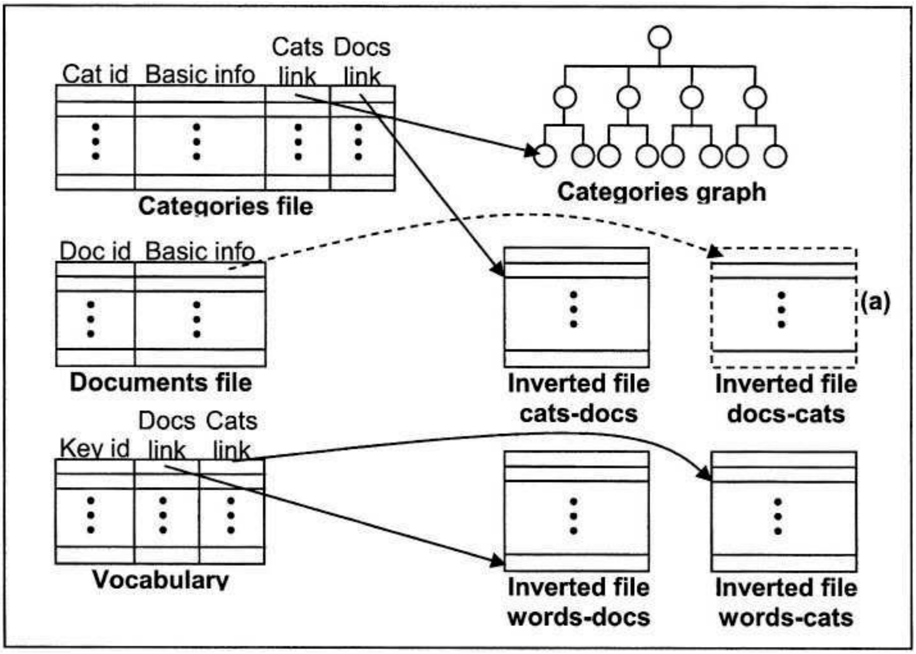


Fig. 1. Data structure of the basic model for a Web directory.

2.1 Restricted Searches

The restricted search process in a DAG area requires a more elaborated access to that information. On the one hand, a standard search is carried out by means of accessing the inverted word file and combining the inverted document lists in the usual way. Once the list of results is calculated, the key step consists of determining which documents belong to the restricted DAG area. From this basic data model, two alternatives are defined for this filtering process.

The first alternative consists of obtaining the list of documents associated to the specified DAG area. Later on, this inverted list is combined with the list of results. The second alternative consists of obtaining the categories list from the restriction area and checking the result list sequentially, deciding which documents are associated with nodes of the category list.

In the former case, obtaining the list of documents included in a specific DAG area is a hard process that could be divided into three steps. Initially the DAG has to be explored until every leaf node is reached, starting from the root node of the restriction area. A list of associated documents must be obtained for each node and all these lists have to be combined. As a result, a list of all the documents associated to the restricted area is obtained. The final step is to intersect this list with the results list in order to obtain the relevant documents associated to the restricted DAG area.

In the latter case, obtaining the category list (sorted by identifier) just requires exploring the restricted area of the DAG, storing the identifiers in a sorted list. But an auxiliary structure indicating the category associated to each document is necessary (see Figure 1 (a)). This entails an inverted file relating each document to the categories it belongs to (this is a redundant index structure relating categories to documents, reciprocal to the index already defined). In this way, the associated categories, for each document of the results list for the basic query, are retrieved (using this auxiliary structure) and compared with the categories list of the restricted area. The documents with no categories in the restricted area are removed from the final result list.

None of these two alternatives is a definitive solution. The main drawback of the first method lies in the excessive combination of lists to obtain the list of documents associated to the restriction area. This is directly related with the performance of inverted files: the combination (union or intersection) of two or more inverted lists worsens the performance [3]. Assuming that inverted lists are stored sorted by identifier (as mentioned before, this could not be the default option in most commercial systems), the performance is inversely proportional to the number of lists and to the number of elements.

The main disadvantage of the second solution is the need for a redundant data structure. However, the required storage space is rather small, since the number of documents indexed in Web directories is rather limited (it is generally assumed that less than 1% of the Web is catalogued), and each document usually belong to a limited number of categories (typically, 1 or 2).

From a global point of view, the first method is adequately adapted to those queries where the restricted DAG area is reduced (which corresponds to the lower DAG levels), given that the number of involved categories (and, therefore, the number of documents to be combined) is small. On the other hand, the second alternative is efficient when the number of results obtained in the search is reduced (regardless of the amplitude of the restriction area), since both the sequential reading and the index access will be moderate.

However, none of the two solutions solves efficiently the searches that obtain a great number of results and that have been restricted to a wide DAG area (the study explained at [5] shows that most searches are restricted to categories in the first three levels).

3 Hybrid Model

The same problem has been previously considered in [6], where a solution based on hybrid data structures was proposed. In this paper the solution described is based on the second alternative of the basic model described, using signature files as an inexact filter that will reject most of the results that are not associated to the restricted categories. Thus, the exact filtering only examines the remaining documents, whose number will have been considerably reduced.

Signature files are an inexact filtering technique (measured through the false drop probability) based on sequential access to data, which constitutes its main disadvantage, producing a poor performance. However, in this case, that is no inconvenient, since the sequential filtering is only carried out within the search results, never with the whole set of documents in the collection. The signature file technique is based on the superimposing codes [14]. In this technique, each document is divided into blocks that contain a constant number of different words. A signature is linked to each word, and the block signature is obtained by means of the bit-to-bit logic OR of its words' signatures. The search is carried out from the signature of the searched word, just by making a logic AND between the word's and the document's signatures, checking whether it coincides with the word's original signature. For more details about signature files and superimposed codes refer to [8, 14, 15].

In this model the superimposing codes have been adapted to the categories DAG environment. Each document must have a signature that represents each and every one of the categories it belongs to, directly or indirectly. The signature files will be incorporated to the inverted files, creating a hybrid scheme of inverted and signature file. A composed document identifier is defined to dynamically generate the signature files associated to each query (merging several inverted lists). This identifier is composed of the superimposed signature of every category a document is associated to (directly or indirectly) and a local document identifier.

With this solution the functioning scheme for the restricted searches is the following:

- First, the list of results is obtained from the standard search process, with no restrictions on their arrangement (normally, according to a relevance ranking).
- Second, the inexact filtering is carried out for the category to which the search is restricted using the signature file associated to the list of results.
- Third, the exact filtering of the rest of results is carried out, according to the second alternative previously described.

Based on this solution two variants are defined: the hybrid model with total information and the hybrid model with partial information. The former corresponds to the direct application of the superimposing code technique to the whole DAG of categories: each and every category has an associated signature. In the latter, the superimposing codes are used only in the upper levels of the DAG (typically, the first three levels) and the rest of nodes will inherit the signatures.

From a general point of view, the main disadvantage of this method lies in the space required by the composed identifier, which will increase the size of the inverted file that relates keywords and documents. In fact, the hybrid model with total information will duplicate the size of this inverted file. The variant with partial information optimises this size reducing in a 65% the increase in the storage space required.

In the performance evaluation, both of the hybrid models increase the performance of the IR system for the restricted queries under void, low and medium workload situations. But in a high workload environment only the variant with partial information is able to keep the improvement over the basic model. This is directly related with the increase in the inverted file size of this solution.

4 Optimistic Model

The main feature of Web directories is that their ontology is a DAG, which provides them with a great cataloguing flexibility: one category may have several parents and one document may belong to several categories.

Based on the study developed in [5], the distribution of categories and documents was analysed for a real Web directory. In this case, BIWE, a Spanish Web directory, with approximately 1000 categories and 50,000 documents. The distribution of categories was the following: 84.3% have only one parent, 14% have two parents and only 1.7% have three or four parents. For the documents, a 72.77% are associated with one category, 20.9% with two categories, 5.15% with three categories and only 1.18% with four or more categories.

These figures lead to the idea of an optimistic model, because the majority of the DAG could be represented in a simple tree. So, in the proposed model the DAG will be handled as a tree with some "exceptions", but continues to be a DAG.

With this purpose, the categories of the DAG are labelled following the route: root, left sub-tree and right sub-tree. If a category has two or more parents only the label of the first route will be considered (for example, see Category 4 on Figure 2). Also, a range is associated to each node, starting in the lowest label (usually the own category) and finishing in the highest label of its descendants. To deal with the exceptions a list of ranges is stored in each node. For example, in Figure 2 category number 5 contains two ranges because one of its descendants has two different parents. In this way, the ranges concentrate all the information about the descendants of a category.

For the documents, a data structure is used to store the categories associated directly with each document. This could be done using an inverted file that associates documents with categories (creating an inverted file redundant with the categories-documents structure, as shown in Figure 1 (a)), or a sequential list could be stored in the documents file, defining a maximum limit for the number of categories associated to each document (see Figure 2).

Also, the documents identifiers are defined using a special code. Each document using this code will be univocally identified and, at the same time, some information about its associated categories will be included. This code will store the information about *N* categories, where *N* will depend on the implementation. Obviously, as more categories are included in the code (larger *N*), more space is required for each document identifier.

Generally, the code is composed of the following fields: a categories counter, id. category *1*, id. category *2*, ..., id. category *N* and a local document id. The *categories counter* represents the number of categories associated with this document. The next *N* fields represent the category identifiers associated with this document (some of the identifiers could be void). The *local document id* is a document identifier relative to all the previous category identifiers. If a document is associated with more than *N* categories, the first *N* categories identifiers could be stored, but some will be missed.

Following with the example of Figure 2, Table 1 describes the values of the document identifiers coded, for all the documents. An implementation is assumed with *N* = *2*. In this way, for the documents associated to one or two categories, using only the document identifier, all their associated categories could be easily retrieved.

Therefore, using the optimistic model described, the restricted search process is reduced to the following steps (assuming that the standard search has been performed and the results have been obtained sorted in any type of relevance ranking):

- First, the ranges of the restricted category are obtained directly in the category file. At this point, it is not necessary to explore the restricted area of the DAG of categories, as in the hybrid model.
- Second, the identifiers of the list of results are examined sequentially to decide whether a document belongs or not to the restricted area. If any of the categories associated with a document is included in the ranges, then the document will classify. If none of the categories identified in the code is included in the ranges and some categories are missed, then the document is undetermined. Otherwise, the document will not classify.
- Third, for all the undetermined documents, it is necessary to access the auxiliary data structure to confirm if the document will classify or not.

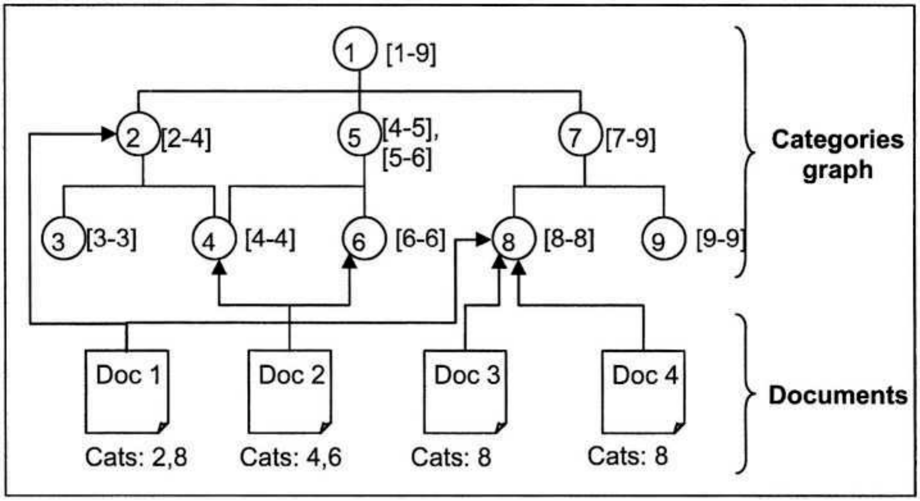


Fig. 2. General description of the optimistic model.

Table 1. Example of document identifier code.

Document	Categories counter	Category 1	Category 2	Local document id
1	2	2	8	1
2	2	4	6	1
3	1	8	-	1
4	1	8	-	2

This process is based on the assumption that the great majority of the documents will be classified in the second point and very few (or even none) will need to enter the third step, which requires more disk accesses and could increase the response time.

The next section describes the implementation details of the optimistic model and the performance evaluation against a basic model and the hybrid model with partial information.

5 Implementation of the Optimistic Model

The implementation that has been carried out consists of developing a Web directory prototype based on a real environment. This prototype consists of a DAG of categories integrated by approximately 900 categories distributed into 7 depth levels, in which more than 51,000 Web pages have been classified. Obviously, there are categories that present several parents and documents associated to multiple categories, in the same percentages as in the real environment (16% and 27% respectively).

The development of the prototypes is based on a three-layer architecture, using Oracle as managing system for the database of the lower layer. Services have been developed by means of Java Servlets, which use several APIs specifically designed for this environment. The development environment has been a Sun Ultra Enterprise 250 machine with a processor at 300 MHz, 768 MB memory and 18 GB storing space.

In this implementation of the optimistic model, the coding scheme previously defined for the document identifiers has been slightly modified to solve approximately 93.5% of the cases directly and reduce the storage space needed. A 32-bit document identifier is used (a reasonable size and 14 bits smaller than the one used in the hybrid model with partial information), and *N* is set to 2. Four fields are defined in the code: the categories counter, two category identifiers and the local document identifier of variable size (see Table 2).

The categories counter is set to 1 if the document belongs to two or more categories and to 0 if it is associated to only one category. If the document belongs to more than two categories, the first category identifier field is set to 1023 (all ones). In this way, only one bit is required for this field (versus the two necessary for the general code defined previously), although no category information is provided for the documents associated with more than two categories.

The next two fields will store the valid category identifiers for the documents that are associated with one or two categories. The last field is a local document identifier relative to each category identifier. Initially, the size of this local document identifier is 20 bits (allowing more than one million documents per category), which is reduced to 10 bits if the document is associated with two categories, due to the use of 10 extra bits by the category identifier number 2. This will limit to 1023 the number of documents associated with the same two categories. In fact, this is a quite reasonable value, since two different categories sharing more than 1000 documents definitively suggests the creation of a new category for these documents.

In the optimistic model also the information about the categories associated to each document must be stored and two main solutions are defined. First, it is possible to store in the document file a list of the categories associated to each document id (probably limiting the maximum number of categories associated to a document). The second alternative consists of storing this information in an inverted file linked from the document file.

In both cases it is possible to optimise the storage size using the code defined, and store only the information about the documents that are associated with three or more categories (in fact only 6.3% of them). This is possible because in the remaining cases the own document identifiers will have this information included.

With this model, there is no increase in the index size because the size of the document identifier remains stable (oppositely as the hybrid models). The number of documents supported by the system is quite high (on average each category could handle one million documents), the main drawback of this model is the limitation in the maximum number of categories.

With the code defined the maximum number of categories is 1023 (starting in the category 0 and with the identifier 1023 reserved). This limit can be easily extended to 2047 or 4095, just reducing the number of documents related with the same two categories from 1024 to 256 or 64, respectively.

A variant of this model can be defined that operates only on the first levels of the DAG. In this case, the limitation would refer only to the categories of the first levels, while the number of low categories could be increased without restrictions. This variant is also supported by the fact that the majority of the restricted queries are centred in the first levels of the DAG, as it is shown in [6]. Obviously, this solution will have a negative impact in the queries restricted to the lower levels that requires further research.

The insertion of a new category in this optimistic model may require the modification of part of the DAG identifiers, which maybe critical for the system performance. This effect could be minimized reserving some identifiers for future uses when the DAG is labelled (for example, for each main branch of the graph).

Table 2. Document identifier code for the optimistic model.

Document associated to	Categories counter	Category id 1	Category id 2	Local document id
1 category	0	10 bits	void	20 bits
2 categories	1	10 bits	10 bits	10 bits
3+ categories	1	10 bits (all 1's)	void	20 bits

5.1 Performance Evaluation

This section's goal is to contrast the response time for the restricted searches, as one of the central parameters established at [17], in the optimistic model against the basic model (defined in Figure 1) and the hybrid model with partial information. The basic model will be the baseline and the improvements in the response time will be relative to this model.

The methodology used for evaluating the performance adapts to the environment of the IR systems in the Web, which are characterised by supporting variable load levels through time. In this way, the performance evaluation under different workload environments provides a more exhaustive analysis of the indexing techniques than just an evaluation in an ideal situation. In particular, it can detect drastic changes in the performance as the workload of the system varies [6].

Five load situations shall be considered for this evaluation: void, low, medium, high and saturation. Determining the saturation point of the prototypes that were developed, allowed to set the various load points. The results were 5 searches per minute for low load, 12 searches for medium load, 19 searches for high load, and 23 for saturation, with the corresponding accesses to categories and documents. The load is generated using a simulation tool especially designed for that purpose. This tool sends requests (queries, browsing categories and visiting documents) to the search system simulating the behaviour of the users. As mentioned before, the IR systems are executed on a Sun Ultra Enterprise 250 and when the tests were performed only the search engines were running.

In each search system and for each workload environment a pool of restricted queries was sent and the response time was timed. The pool contained standard queries (derived from real queries to a Web directory, based on the work in [5]) that retrieved from 0 to 2000 different documents, which were restricted to different areas of the DAG of categories. The queries were randomly sent to the system and this process was repeated 5 times.

An ANOVA test was carried out to determine the influence of each model (basic, hybrid with partial information and optimistic). Three factors have been taken into account: model, number of results and number of documents in the restricted area. The number of results and the number of documents in the restricted area are parameters of great influence in the response time and the goal is to determine if the type of model is also significant.

Very similar results were obtained for the void, low, medium and high loads. Figures 3, 4, 5 and 6, respectively show the average response time (in milliseconds) for each model, according to the number of search results.

As may be seen in these figures, the hybrid model and the optimistic model greatly improve the performance of the basic model in all the workload situations. This improvement is clearer when the number of results retrieved in the standard search is high (over 500 results), since the performance of the basic model falls dramatically. As mentioned before, the basic model will produce its worst results in this type of queries, although in some queries the response time could improve, for example because the number of results (of the global query) is small. The hybrid and optimistic models reduce the response time in approximately 50% over the basic model. Besides, the ANOVA test considers that the type of model is a significant parameter with a 0.999 probability and all the factors analysed explain more than 91.6% of the variability of the data (in all the cases).

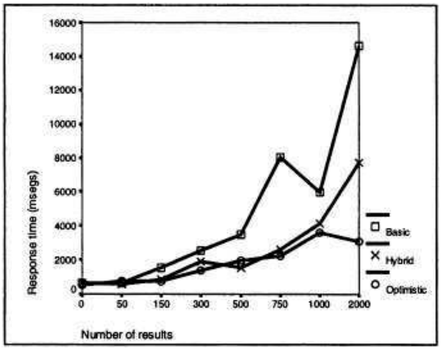


Fig. 3. Estimated response time according to the number of results (void load).

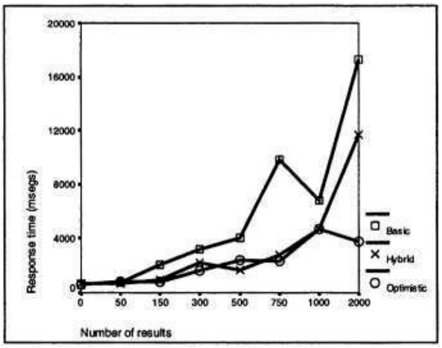


Fig. 4. Estimated response time according to the number of results (low load).

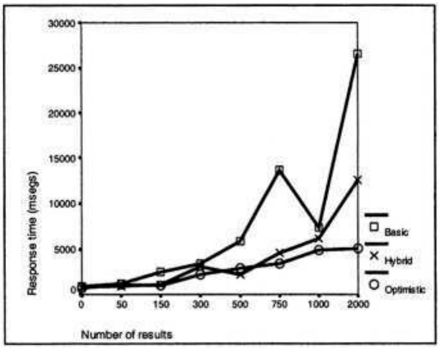


Fig. 5. Estimated response time according to the number of results (medium load).

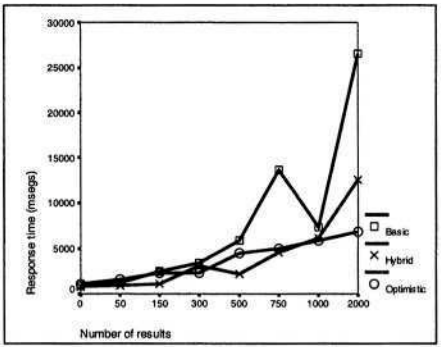


Fig. 6. Estimated response time according to the number of results (high load).

Table 3. Estimated mean response time (milliseconds) for the query that retrieve 2000 documents in all the workload environments.

Workload	Hybrid	Optimistic	% Improvement
Void	7725	3039	60.66%
Low	11705	3786	67.65%
Medium	12609	5071	59.78%
High	13234	6837	48.34%
Saturation	31812	10160	68.06%

The main difference among the different workloads lies in a generalised increase in the response time when the workload of the IR system increases.

In addition there is an important difference between the hybrid model and the optimistic model that can be seen when the number of results retrieved is large enough (in our case, when the standard queries retrieve 2000 documents). In all the graphs both models perform more or less similarly, but on the final step the optimistic model always outperforms the hybrid model. As it is shown in Table 3, the optimistic model improves the average response time of the hybrid model (when the queries retrieve 2000 documents) in more than a 61%.

The better performance of the optimistic model is due to the filtering process. In the hybrid model the inexact filter will remove a lot of documents that will not classify, but it is necessary to check all the remaining ones using an exact filtering. This is because the superimposing codes will include some "false drops" in the final results that need to be removed.

On the other side, the optimistic model and the code used in our implementation (see Table 2) will perform a first filtering process that will return all the documents that will classify and some documents that could (or not) classify. In this case, it is necessary to check only the doubtful documents. As was previously mentioned, our implementation only will produce, on average, 6.3% of doubtful documents that will need to be reviewed. On the contrary, in the hybrid model 100% of the documents retrieved in the first filtering process are doubtful and will have to be reviewed. This point makes the difference in the response time obtained, especially when the number of documents retrieved is high enough to obtain a high number of restricted documents.

Figure 7 shows another interesting point, where the response time has been obtained in a saturated environment. In this picture it is clear that the optimistic model outperforms both, the basic and the hybrid model in nearly all the restricted queries. The improvement over the basic model is related with the optimisation of the restricted search process that reduces the number of disk accesses. The difference between the hybrid and the optimistic model is due to the size of the search index, because the optimistic model uses 32-bit document identifiers versus the 46-bit identifiers of the hybrid model. This difference is clear when the disk operations are highly demanding and in this case, the system that requires less disk access will perform better, due to caching and locality of reference.

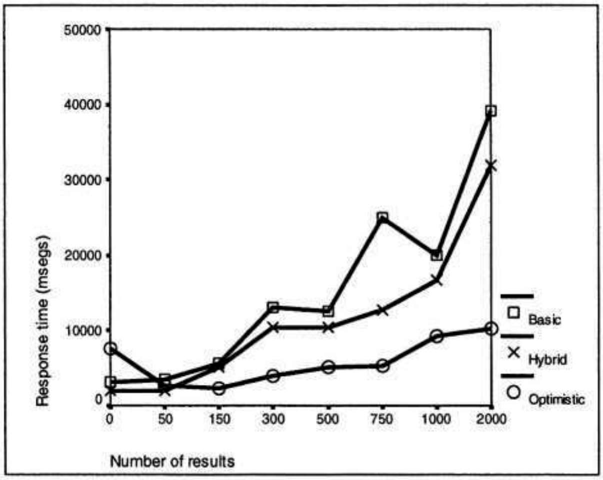


Fig. 7. Estimated response time according to the number of results (saturation load).

6 Conclusions

This paper describes a data structure named optimistic model especially designed to improve the performance of the restricted searches in a Web directory. The proposed model was compared with a basic model and a hybrid model (with partial information).

From the design and implementation point of view, the optimistic model constitutes a simpler data structure than the hybrid model. The hybrid model needs to estimate the best values for the parameters of the signature files used, which can lead to future troubles if the distribution of categories and documents in the Web directory changes.

On the other side, the proposed model presents some restrictions in the number of categories supported by the system, due to the code scheme used. In our case, this limitation could be easily doubled or quadrupled just changing slightly the code used, although some other solutions should be studied in the future.

From a general point of view, the optimistic model is able to improve 50% the response time of a basic model. With respect to the hybrid model, both systems provide similar response time, except in the case of large answers. In this case, the optimistic model outperforms the hybrid model in approximately 61%.

In the saturation workload environment the optimistic model proved to perform better than the basic model and the hybrid models for all types of queries. The proposed model will operate using smaller document identifiers than the hybrid model and this will reduce the size of the search index, remaining stable with regard to the baseline. This point is especially important when the number of disk operations is highly intensive.

Future research lines will be devoted to improve the flexibility of the proposed system with respect to the number of categories. Related with this point, the use of compression techniques in the code scheme used should be explored. Also, the relation between the distribution of the number of categories with several parents and the system performance is a key point for future works.

The variation proposed over the general optimistic model that operates only on the categories of the first levels of the DAG. This new model has to be analysed deeply, but a reduction in the limitation of the number of categories could be easily achieved. Another extension of this work is the use of lazy evaluation to process the multiple lists used in a restricted query. In this way, the processing is done only when it is finally required, and some reductions in the response time could be achieved.

Our results could also be applicable to similar restricted searches. For example, instead of a category DAG we may have a domain tree and restrict searches to a site or a country in a Web search engine.

Acknowledgements. The work of the first author has been partially supported by the Spanish CICYT, under project TIC2001-0547, and that of the second author by CYTED project RIBIDI.

References

1. M. Agosti, M. Melucci. “Information Retrieval on the Web”, in: M. Agosti, F. Crestani, G. Pasi (Eds). Lectures on Information Retrieval: Third European Summer School, ESSIR 2000. Revised Lectures, Springer-Verlag, Berlin/Heidelberg, 2001, 242-285.
2. R. Baeza-Yates, “Searching the Web”. In R. Baeza-Yates and B. Ribeiro-Neto, “Modern Information Retrieval”, chapter 13, pp: 367-395. Addison Wesley, 1999.
3. R. Baeza Yates, G. Navarro, “Indexing and Searching”. In R. Baeza-Yates and B. Ribeiro-Neto, “Modern Information Retrieval”, chapter 8, pp 191-228. Addison Wesley, 1999.
4. S. Brin, L. Page, “The anatomy of a large-scale hypertextual web search engine”. The 7th International World Wide Web Conference, 1998.
5. F. Cacheda, A. Viña, “Experiencies retrieving information in the World Wide Web”. 6th IEEE Symposium on Computers and Communications, pp 72-79, 2001.
6. F. Cacheda, A. Viña, “Optimization of Restricted Searches in Web Directories Using Hybrid Data Structures”. In 25th European Conference on Information Retrieval Research (ECIR’03), Lecture Notes on Computer Science (2633), pp. 436-451, 2003.
7. D. Cutting, J. Pedersen, “Optimizations for dynamic inverted index maintenance”. 13th International Conference on Research and Development in Information Retrieval, 1990.
8. C. Faloutsos, S. Christodoulakis, “Description and performance analysis of signature file methods”. ACM TOOIS, 5 (3), 237-257, 1987.
9. Google, http://www.google.com/, 2003.
10. D. Harman, E. Fox, R. Baeza-Yates, W. Lee, “Inverted files”. In W. Frakes and R. Baeza-Yates, “Information Retrieval: Data structures and algorithms, chapter 3, pp: 28-43. Prentice-Hall, 1992.
11. G. Jacobson, B. Krishnamurthy, D. Srivastava, D. Suciu, “Focusing Search in Hierarchical Structures with Directory Sets”. Seventh International Conference on Information and Knowledge Management (CIKM), 1998.
12. Y. Labrou, T. Finin, “Yahoo! as an ontology – Using Yahoo! categories to describe documents”. Eighth International Conference on Information Knowledge Management (CIKM), pp. 180-187, 1999.
13. The Open Directory Project, http://www.dmoz.org/, 2003.
14. C. S. Roberts, “Partial-match retrieval via the method of superimposed codes”. Proceedings of the IEEE, 67:12, 1624-1642, 1979.
15. S. Stiassny, “Mathematical analysis of various superimposed coding methods”. American Documentation, 11 (2), 155-169, 1960.
16. Yahoo!, http://www.yahoo.com/, 2003.
17. J. Zobel, A. Moffat, K. Ramamohanarao, “Guidelines for Presentation and Comparison of Indexing Techniques”. ACM SIGMOD Record, 25(3):10-15, 1996.
18. J. Zobel, A. Moffat, K. Ramamohanarao, “Inverted files versus signature files for text indexing”. ACM Transactions on Database Systems, 23(4), pp.453-490, 1998.

Content-Aware DataGuides: Interleaving IR and DB Indexing Techniques for Efficient Retrieval of Textual XML Data

Felix Weigel[1], Holger Meuss[2], François Bry[1], and Klaus U. Schulz[3]

[1] Institute for Computer Science, University of Munich, Germany
[2] European Southern Observatory, Garching, Germany
[3] Centre for Information and Language Processing, University of Munich, Germany

Abstract. Not only since the advent of XML, many applications call for efficient structured document retrieval, challenging both Information Retrieval (IR) and database (DB) research. Most approaches combining indexing techniques from both fields still separate path and content matching, merging the hits in an expensive join. This paper shows that retrieval is significantly accelerated by processing text and structure simultaneously. The *Content-Aware DataGuide (CADG)* interleaves IR and DB indexing techniques to minimize path matching and suppress joins at query time, also saving needless I/O operations during retrieval. Extensive experiments prove the CADG to outperform the DataGuide [11,14] by a factor 5 to 200 on average. For structurally unselective queries, it is over 400 times faster than the DataGuide. The best results were achieved on large collections of heterogeneously structured textual documents.

1 Introduction

Many modern applications produce and process large amounts of semi-structured data which must be queried with both structural and textual selection criteria. The W3C's working drafts supporting full-text search in XQuery and XPath [4, 1] illustrate the trend towards the integration of structure- and content-based retrieval [2]. Consider e.g. searching a digital library or archive for papers with, say, a title mentioning *"XML"* and a section about *"SGML"* in the related work part. Obviously, the query keywords *("XML", "SGML")* as well as the given structural hints (title, related work) are needed to retrieve relevant papers: searching for *"XML"* and *"SGML"* alone yields many unwanted papers dealing mainly with SGML, whereas a query for all publications with a title and related work possibly selects all papers in the library. The same holds for retrieval in structured web pages or manuals, tagged linguistic or juridical corpora, compilations of annotated monitoring output in informatics or astronomy, e-business catalogues, or web service descriptions. Novel Semantic Web applications will further increase the need for efficient structured document retrieval. All these applications (1) query semi-structured data which (2) contains large text portions and (3) needs persistent indices for both content and structure.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 378–393, 2004.

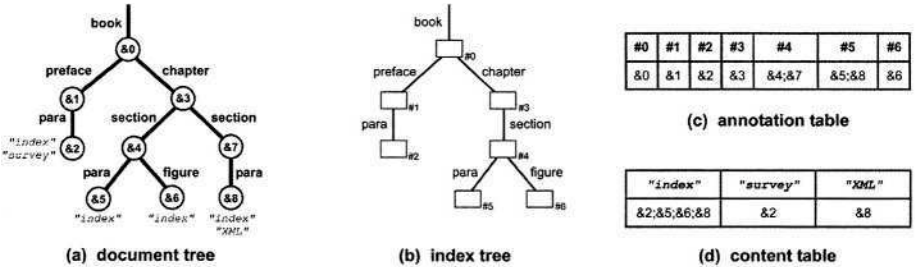
Ongoing research addresses this characteristic class of data by combining data structures from Information Retrieval (IR) and database (DB) research. Common IR techniques for flat text data are *inverted files* [9,24] and *signature files* [9,8]. The ground-breaking DB approach to indexing semi-structured data is the *DataGuide* [11,14], a compact summary of label paths. A mere structure index, it is used with an inverted file for text in [14], but has also been combined with signature files [5] and Tries [6] (see section 6). In these approaches structure and content are still separated and handled sequentially, usually requiring an expensive join at query time. We show that both inverted files and signatures can be tightly integrated with the DataGuide for simultaneous processing of structure and text in all retrieval phases. At the same time, part of the content information is shifted to main memory to minimize retrieval I/O. This improves performance by a factor 5 to 200 on average, and up to 636 under favourable, yet realistic conditions, namely few structural constraints but rather selective keywords. Excelling in poorly structured queries, our approach meets common demands, since users rarely explore the document schema before querying and tend to focus on content rather than structure, accustomed to flat-text web search engines.

This paper is organized as follows. Section 2 describes the DataGuide and a simple query formalism. Section 3 explains approaches to *content awareness,* one of which (the *structure-centric* one) is shown to be superior. Section 4 presents two structure-centric Content-Aware DataGuides, *Inverted File CADG* and *Signature CADG.* Section 5 reports on the experimental comparison of both with the DataGuide. The paper concludes with related work and future research.

2 Indexing XML with the Original DataGuide

A well-known and influential DB approach to indexing semi-structured data (in this paper, we focus on tree-shaped XML data) is the *DataGuide* [11,14], a summary of the document tree in which all distinct label paths appear exactly once. Figure 1 *(b)* shows the DataGuide for the document tree in *(a).* Both differ in that multiple instances of the same document label path, like /book/chapter/section in *(a),* collapse to form a single index label path *(b).* Hence the resulting *index tree,* which serves as a path index, is usually much smaller than the document tree (though theoretically it can grow to the same size). Thus it typically resides in main memory even for large corpora.

Without references to individual document nodes, however, the index tree only allows to find out about the existence of a given label path, but not its position in the corpus. To this end, every index node is *annotated* with the IDs of those document nodes reached by the same label path as itself. For instance, the index node #4 in figure 1 *(b)* with the label path /book/chapter/section points to the document nodes &4 and &7. The annotations of all index nodes are stored on disk in an *annotation table (c).* Formally, it represents a mapping $dg_a : i \mapsto D_i$ where i is an index node ID and D_i the set of document nodes reached by i's label path. Together index tree and annotation table encode nearly the whole structure of the document tree (except parent/child node pairs).



(a) document tree

(b) index tree

#0	#1	#2	#3	#4	#5	#6
&0	&1	&2	&3	&4;&7	&5;&8	&6

(c) annotation table

"index"	"survey"	"XML"
&2;&5;&6;&8	&2	&8

(d) content table

Fig. 1. Data structures of the original DataGuide

The DataGuide, as a path index, ignores textual content. In [14], it is combined with an inverted file mapping any keyword k to the set D_k of document nodes where k occurs, $dg_c : k \mapsto D_k$. The file resides in a separate *content table* on disk, as figure 1 *(d)* shows. (Non-first normal form (NF^2) is not mandatory.)

In our tree query language [15], there are *structural* query nodes, matched by document nodes with a suitable label path, and *textual* ones containing keywords. A query path consists of structural nodes, linked by labelled edges, and possibly a single textual leaf holding a con- or disjunction of query keywords (see figure 2). Edges to structural query nodes may be either *rigid* (solid line), corresponding to XPath's `child` axis, or *soft* (dashed line, `descendant`). Similarly, if a textual query node has a rigid edge, its keywords must occur directly in a document node matching the parent query node. With a soft edge, they may be nested deeper in the node's subtree. Query processing with the DataGuide requires four *retrieval phases:*

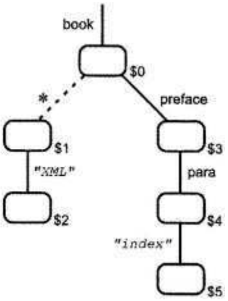


Fig. 2. Query tree

1. *Path matching:* the query paths are matched separately in the index tree.
2. *Occurrence fetching:* annotations for index nodes found in phase 1 are fetched from the annotation table; query keywords are looked up in the content table.
3. *Content/structure join:* for each query path, the sets of annotations and keyword occurrences are joined (i.e. in the easiest case, intersected).
4. *Path join:* path results are combined to hits matching the entire query tree.

While phases 1 and 3 take place in main memory, phase 2 involves at least two disk accesses. Phase 4 may, but need not, require further I/O operations. The following example shows drawbacks of separate structure and content matching.

Example. Consider the query `/book//*["XML"]` (left path in figure 2), selecting all document nodes below a book root which contain the keyword *"XML"*. In phase 1, the query path /book//* is searched in the index tree from figure 1 *(b)*. All index nodes except the root qualify as structural hits. In phase 2, fetching the annotations of the index nodes #1 to #6 in *(c)* yields the six annotation sets

{&1}, {&2}, {&3}, {&4;&7}, {&5;&8}, and {&6}. Obviously unselective query paths featuring // and $*$ may cause multiple index nodes to be retrieved during path matching. According to *(d)* the occurrence set of *"XML"* is {&8}. In phase 3, it is intersected with each annotation set to find out which of the structural matches contain the query keyword. Here almost all candidate index nodes are discarded, their respective annotation set and occurrence set being disjoint. Only #5 references a document node meeting both criteria. The document node in the intersection $\{\&5; \&8\} \cap \{\&8\} = \{\&8\}$ is returned as the only hit for the query node $1. The second path in figure 2 is processed analogously. Path joining (phase 4) is omitted here. Soft-edged textual nodes entail exhaustive index navigation and a more complex join (see [21] for details and optimizations).

In the above example many false positives (all index nodes except #5) survive path matching and fetching, to be finally ruled out in retrieval phase 3. Not only does this make phase 1 unnecessarily complex; it also causes needless I/O in phase 2. Since structural and textual selection criteria are satisfied when considered in isolation, only phase 3 reveals the mismatch. Note that a reverse matching order – first keyword fetching, then navigation and annotation fetching – has no effect, unless keyword fetching fails altogether (in which case navigation is useless, and the query can be rejected as unsatisfiable right away). Moreover, it results in similar deficiencies for queries with selective paths, but unselective keywords. In other words, the DataGuide faces an inherent defect, keeping structural and textual selection criteria apart during phases 1 and 2. We propose a *Content-Aware DataGuide* which combines structure and content matching from the very beginning of the retrieval process. This accelerates the evaluation process especially in case of selective keywords and unselective paths.

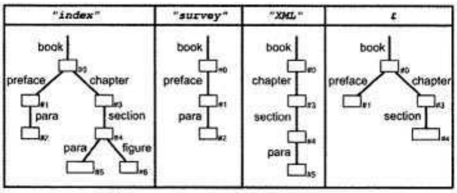
3 Two Approaches towards Content Awareness

Content awareness strives to exploit keyword information during both path matching and annotation fetching in order to suppress needless I/O and joins at query time. It enhances the DataGuide with a materialized content/structure join and a keyword-driven path matching procedure. We propose an exact and a heuristic technique to prune index paths which are irrelevant to a given set of query keywords. This *content-aware navigation* not only reduces the number of paths to be visited, but also excludes false positives from annotation fetching. Precomputing the content/structure join at indexing time allows document nodes to be retrieved simultaneously by their label path and content, avoiding the intersection of possibly large node ID sets at query time. A single *content-aware annotation fetching* step replaces the two table look-ups in phase 2.

We examine two symmetric approaches to meeting the above objectives. The *content-centric approach* (see section 3.1), being simple but inefficient, only serves as starting point for the more sophisticated *structure-centric approach,* which is pursued in the sequel. Section 3.2 presents it from an abstract point of view. Two concrete realizations, as mentioned above, are covered in section 4.

3.1 Naive Content-Centric Approach

One way of restricting path matching to relevant index nodes is to use multiple index trees, each covering the document paths with a specific keyword, as in figure 3 *(a)*. Similarly, annotations are grouped by keyword to materialize the content/structure join in phase 3: replacing the DataGuide tables, a *content/annotation table (b)* maps a keyword k and an index node ID i to the set $D_{k,i}$ of i's annotations containing k, $cadg_{cc} : (k, i) \mapsto D_{k,i}$. Obviously the content/annotation table easily takes up more space on disk than the DataGuide's tables. Its size increases with the number of keywords occurring under many different label paths. This storage overhead is subject to a time/space trade-off common to most indexing techniques. The main drawback of the content-centric approach is that not only the content/annotation table but also the typically large and redundant set of index trees must reside on disk. For each query keyword the right tree needs to be loaded at query time.



(a) content-centric index trees

"index"			"survey"	"XML"	ε			
#2	#5	#6	#2	#5	#0	#1	#3	#4
&2	&5;&8	&6	&2	&8	&0	&1	&3	&4;&7

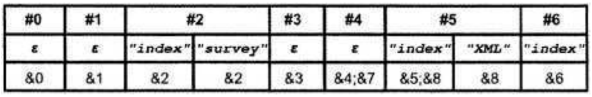
(b) content-centric content/annotation table

Fig. 3. Content-centric CADG

3.2 Structure-Centric Approach

The approach just presented is *content-centric* in the sense that keywords determine the structure of the index tree and table. A more viable approach to content awareness preserves the tree in its integrity, grouping the keyword occurrences by their label paths. This *structure-centric* approach allows path matching to be performed without loading index trees from disk. The tree resides in main memory like the one of the DataGuide. It resembles figure 1 *(b)*, except that each index node holds enough content information to prune irrelevant paths during phase 1. Dedicated data structures to be presented in the next section encode (1) whether an index node i references any document node where a given keyword k occurs, and (2) whether any of i's descendants (including i itself) does. Henceforth we refer to the former relation between i and k as *containment* and to the latter as *government*. While government is examined for any index node reached during phase 1 in what we call the *government test*, a *containment test* takes place in phase 2. Both are integrated with the DataGuide's retrieval procedure to enable content-aware navigation and annotation fetching, as follows. In phase 1, whenever an index node i is being matched against a structural query node q_s, the procedure $governs(i, q_s)$ is called. It succeeds iff for each textual query node q_t below q_s containing a keyword conjunction $\bigwedge_{u=0}^{p} k_u$, condition (2) above is true for i and all keywords k_u (at least one in case of a disjunction $\bigvee_{u=0}^{p} k_u$). If so, path matching continues with i's descendants; otherwise

i is discarded with its entire subtree. In phase 2, before fetching the annotations of any index node i matching the parent of a textual query node q_t, the procedure $contains(i, q_t)$ is called to verify condition (1) for i and all of q_t's keywords (at least one in case of a disjunction). If it succeeds, i's annotations are fetched; otherwise i is ignored. The realization of *governs*() and *contains*() depends on how content is represented in index nodes—but also in query nodes, which must hint at the keywords below them to avoid repeated exhaustive query tree scans. To this end, suitable *query preprocessing* takes place in a new retrieval phase 0 (see below). Keyword occurrences and annotations are combined in a *content/annotation table* as in figure 4. It can be considered as consisting of seven index-node specific content tables (#0 to #6), each built over a label-path specific view of the data. In first normal form, it is identical to the one in figure 3 *(b)*. Index node and keyword together make up the primary key, enabling content/structure queries as well as pure structure or pure content queries.



#0	#1	#2		#3	#4	#5		#6
ε	ε	"index"	"survey"	ε	ε	"index"	"XML"	"index"
&0	&1	&2	&2	&3	&4;&7	&5;&8	&8	&6

Fig. 4. Structure-centric content/annotation table

4 Two Realizations of the Structure-Centric Approach: Inverted File CADG and Signature CADG

The concept of a structure-centric CADG does not specify data structures or algorithms for integrating content information with the index and query trees. This section proposes two alternative content representations inspired from IR, along with suitable query preprocessing and government/containment tests. The first approach, which uses inverted files (see section 4.1), is guaranteed to exclude all irrelevant index nodes from path matching and annotation fetching. The signature-based CADG (see section 4.2) represents keywords approximately, possibly missing some irrelevant index nodes. A final verification, performed simultaneously with annotation fetching, eventually rules out false positives. Thus both CADGs produce exact results, no matter if their content awareness is heuristic.

4.1 Inverted File CADG (ICADG)

The *Inverted File CADG (ICADG)* relies on inverted index node ID files to enable content awareness. The idea is to prepare a list of *relevant index nodes* for each path in the query tree, comprising the IDs of all index nodes which *contain* the query keywords of this path. Assembled in a query preprocessing step (retrieval phase 0), these lists are attached to the query tree (see figure 5). The index tree lacks explicit content information, just like the DataGuide in figure 1 *(b)*. During retrieval phase 1, only ancestors of relevant index nodes are examined, whereas other nodes are pruned off. Similarly, only annotations of relevant index nodes are fetched during phase 2. Ancestorship among index nodes is tested by navigating upwards in the index tree (which requires a single backlink per node) or else computationally, by means of numbering schemes [13, 20].

Query preprocessing. In retrieval phase 0, each node q in a given query tree is assigned a set I_q of sets of relevant index nodes, as shown in figure 5. Any textual query node q_t with a single keyword k_0 is associated with the set I_{k_0} of index nodes containing k_0, i.e. $I_{q_t} := \{I_{k_0}\}$. I_{k_0} consists of all index nodes paired with k_0 in the content/annotation table (see figure 4). If the query node holds a conjunction $\bigwedge_{u=0}^{p} k_u$ of keywords, their respective sets I_{k_u} are intersected, $I_{q_t} := \{\bigcap_{u=0}^{p} I_{k_u}\}$, because the conjoined keywords must all occur in the same document node, and hence be referenced by the same index node. Analogously, a query node representing a keyword disjunction $\bigvee_{u=0}^{p} k_u$ is associated with the union $I_{q_t} := \{\bigcup_{u=0}^{p} I_{k_u}\}$ of sets of relevant index nodes. If $I_{q_t} = \{\emptyset\}$, the query is immediately rejected as unsatisfiable (without entering retrieval phase 1), because no index node references document nodes with the right content. A structural query node q_s inherits sets of relevant index nodes (contained in a second-order set I_{q_v}) from each of its children q_v $(0 \leq v \leq m)$, $I_{q_s} := \bigcup_{v=0}^{m} I_{q_v}$. Thus the textual context of a whole query subtree is taken into account while matching any single query path. It is crucial to keep the member sets of all I_{q_v} separate in the higher-order set I_{q_s}, rather than intersect them like with keyword conjunctions. After all the children q_v are not required to be all matched by the same document node, which contains occurrences of all their query keywords at once. Hence the government test for an index node i matching q_s must succeed if there exists for each child query node q_v one descendant of i containing the keywords below q_v, without demanding that it be the same for all q_v. For a childless q_s, $I_{q_s} := \emptyset$ is used as a "don't care" to make the government test for q_s succeed (see below). Note that since all children q_v, whether structural or textual, are treated alike the preprocessing procedure also copes with mixed-content queries. Figure 5 depicts the preprocessed query tree from figure 2. Each query node q contains the member sets of I_q (one per row), e.g. $I_{\$0} = \{\{\#5\}; \{\#2; \#5; \#6\}\}$. All ID sets were computed using the content/annotation table in figure 4.

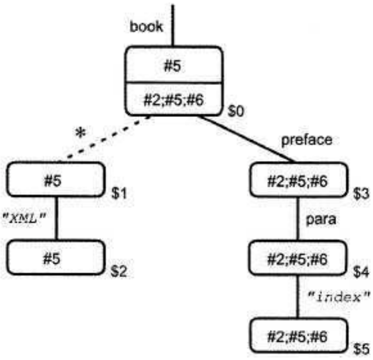


Fig. 5. ICADG query

Government/containment tests. As described in section 3.2, $governs(i, q_s)$ is performed when matching an index node i to a structural query node q_s in phase 1. In each set $J \in I_{q_s}$, a descendant of i (including i itself) is searched.

The test succeeds iff there is at least one in each J. As a special case, this is true for $I_{q_s} = \emptyset$. In phase 2, $contains(i, q_t)$ tests every index node i matching the parent of a textual query node q_t, provided its government test succeeded. This ensures that i or any of its descendants reference a document node containing q_t's keywords. To determine if annotation fetching for i is justified, the index node is searched in the only member set $J \in I_{q_t}$. The test succeeds iff $i \in J$.

Example. Consider the query tree in figure 5 and the DataGuide in figure 1 *(b),* which is identical to the ICADG tree. Its content/annotation table is given in figure 4. After the index root #0 matches the query node $0, *governs*(#0, $0) succeeds because in both sets associated with $0, {#5} and {#2; #5; #6}, there is a descendant of #0 (namely #5). The two paths leaving $0 are processed one after the other. Reached by a soft edge without label constraint, $0's left child $1 is matched by all index nodes except #0. First, *governs*(#1,$1) fails since none of #1's descendants is in $1's list. This excludes the whole left branch of the index tree from further processing. #2 never enters path matching, let alone annotation fetching. #3, as an ancestor of #5, passes the government test for $1, but fails in the containment test for $2 since #3 ∉ {#5}. Its child #4 satisfies *governs*(#4, $1) and fails *contains*(#4, $2) for the same reason as #3. By contrast, #5 passes both tests, being a member of $1's and $2's ID list. Hence its only occurrence of *"XML"*, &8, is fetched from the content/annotation table. #6 is dismissed by *governs*(#6, $1). The second query path is processed analogously.

The above query shows how content awareness saves both main-memory and disk operations. Compared to the DataGuide, two subtrees are pruned during phase 1, and only one disk access is performed instead of seven during phase 2. Another one is needed in phase 0 for query preprocessing (for queries with non-existent keywords, it saves the whole evaluation). The results are identical.

4.2 Signature CADG (SCADG)

Unlike the ICADG, the *Signature CADG (SCADG)* uses only approximate keyword information for path pruning. The resulting heuristic government and containment tests may overlook some irrelevant index nodes, recognizing them as false hits only at the content/annotation table look-up in phase 2. (Nevertheless the retrieval is exact, as explained below.) The content information is added not only to the query tree, but also to the index tree, in the form of *signatures* created at indexing time. Since signatures summarize the content information of entire index subtrees, no ancestor/descendant check is needed for the government test.

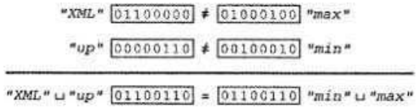

"XML" 01100000 ≠ 01000100 "max"
"up" 00000110 ≠ 00100010 "min"
"XML" ⊔ "up" 01100110 = 01100110 "min" ⊔ "max"

Fig. 6. Ambiguous signatures

Signatures. i.e. fixed-length bit strings, are a common IR technique for concise storage and fast processing of keywords. Every keyword to be indexed or queried is assigned a (preferably unique and sparse) signature. Note that this does not require all keywords to be known in advance, nor to be assigned a signature before indexing. It can be created on the fly, e.g. from a hash code of the character sequence. Sets of keywords in document or query nodes are represented collectively in a single signature, by bitwise disjunction ($\sqcup$) of the individual keyword signatures. As figure 6 shows, overlapping bit patterns may cause ambiguities. Other operations on signatures s_0, s_1 include the bitwise conjunction ($s_0 \sqcap s_1$), inversion ($\neg s_0$), and implication ($s_0 \sqsubset s_1 := (\neg s_0) \sqcup s_1$).

Index tree. The Signature CADG's tree (see figure 7) closely resembles the one of the DataGuide (see figure 1), except that each index node i has two signatures. A *containment signature* s_i^c results from disjoining the signatures of all keywords in i's annotations. (If there is none, s_i^c is $\boxed{00000000}$.) A *government signature* s_i^g encodes keywords referenced by i or a descendant. Inner index nodes obtain it by disjoining the government signatures of their children and their own containment signature. For leaf nodes, s_i^g and s_i^c are identical.

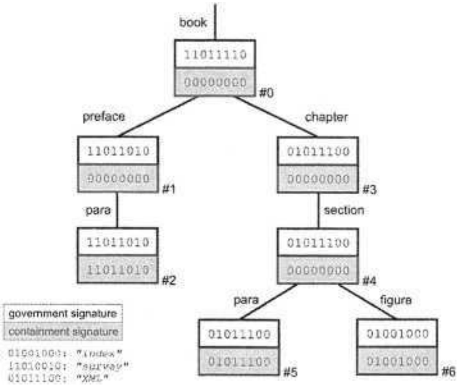


Fig. 7. SCADG index tree

Query preprocessing. Unlike index nodes, every query node q has a single signature s_q. For a textual node q_t, s_{q_t} is created from the signatures s_{k_u} of q_t's keywords k_u $(0 \leq u \leq p)$. If k_0 is the only keyword, then $s_{q_t} := s_{k_0}$. For a keyword *conjunction* $\bigwedge_{u=0}^{p} k_u$, $s_{q_t} := \bigsqcup_{u=0}^{p} s_{k_u}$ is the *disjunction* of the keyword signatures (each k_u "leaves its footprint" in s_{q_t}), whereas $s_{q_t} := \bigsqcup_{u=0}^{p} s_{k_u}$ for a disjunction $\bigvee_{u=0}^{p} k_u$ in q_t. A structural query node's signature s_{q_s} superimposes the child signatures s_{q_v} $(0 \leq v \leq m)$, $s_{q_s} := \bigsqcup_{v=0}^{m} s_{q_v}$, to summarize the textual content of the whole subtree below q_s. If childless, q_s is given $\boxed{00000000}$ to make any index node's government test succeed, as one would expect of a query node without textual constraints. Figure 8 shows the tree query from figure 5, preprocessed for the SCADG. Keyword signatures have been taken from figure 7. They are either created on the fly, or fetched from a *signature table* assembled at indexing time. In our experiments this proved faster despite an additional I/O operation at query time and caused only negligible storage overhead.

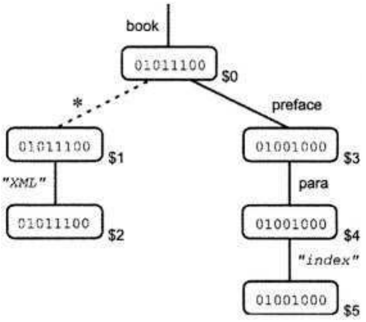


Fig. 8. SCADG query

Government/containment tests. The test $governs(i, q_s)$ for an index node i and a structural query node q_s is $s_{q_s} \sqsubset s_i^g$, requiring all bits set in s_{q_s} to be also set in s_i^g. This holds when q_s's keywords are governed by i. Yet the converse is not always true: as in figure 6, keywords not covered by s_{q_s} can make s_i^g look as if i were relevant to q_s, and path matching continues below i though its descendants must fail in phase 2. At any rate only irrelevant subtrees are pruned. Analogously, the containment test $contains(i, q_t)$ for a textual query node q_t is $s_{q_t} \sqsubset s_i^c$. It succeeds if, but not only if, q_t's keywords occur in i's annotations.

Example. Reconsider the query in figure 8 and the index tree from figure 7 with the content/annotation table in figure 4. After $governs(\#0, \$0) = \boxed{01011100} \sqsubset \boxed{11011110}$ succeeds, path matching continues with query node \$1, matched by all index nodes except #0. Since $governs(\#1, \$1) = \boxed{01011100} \sqsubset \boxed{11011010}$ fails, the antepenultimate bit being set only in \$1's signature, the left index branch is pruned and phase 1 continues with #3. It passes $governs(\#3, \$1) = \boxed{01011100} \sqsubset \boxed{01011100}$, but $contains(\#3, \$2) = \boxed{01011100} \sqsubset \boxed{00000000}$ fails. The same is true for #3's only child #4. While #5 passes *governs* (#5, \$1) (see *governs*(#3,\$1)) and $contains(\#5, \$2) = \boxed{01011100} \sqsubset \boxed{01011100}$, contributing &8 to the result, $governs(\#6, \$1) = \boxed{01011100} \sqsubset \boxed{01001000}$ fails (bits 4 and 6 are missing in $s_{\#6}^g$), saving a table look-up. The second query path is processed analogously.

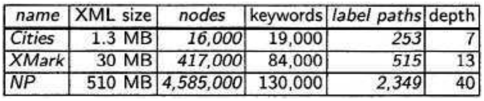
The number of disk accesses compared to the DataGuide is reduced from seven to two (including query preprocessing), like with the ICADG. Moreover, signatures are more efficient data structures than node ID sets (in terms of both storage and processing time) and make relevance checks and preprocessing easier to implement. Note, however, that if another keyword with a suitable signature occurred in #6's annotation, e.g. the keyword *"query"* with the signature $\boxed{00011100}$, then #6 would be mistaken as relevant for \$5's query keyword *"XML"*. The reason is that both $s_{\#6}^g$ and $s_{\#6}^c$, superimposing the *"index"* and *"query"* signatures, would equal the keyword signature for *"XML"*, $\boxed{01001000} \sqcup \boxed{00011100} = \boxed{01011100}$. Hence *governs*(#6,\$1) and *contains*(#6, \$2) would succeed. Only in phase 2 would #6 turn out to be a false hit. This illustrates how the SCADG trades off pruning precision against navigation efficiency.

5 Experimental Evaluation

5.1 Experimental Set-Up

This section describes a part of our experiments with both CADGs and the DataGuide as control. A more detailed report is given in [21]. For the SCADG, we stored 64-bit keyword signatures in a *signature table* at indexing time, created from hash codes with 3 bits set in each signature ([8] surveys signature creation). Extensive tests have been performed on three XML tree corpora with different characteristics (see table 1): *Cities* is small, homogeneous, and non-recursive, whereas *XMark,* a medium-sized synthetically generated corpus [23], is slightly more heterogeneous and contains recursive paths. The highly recursive and heterogeneous *NP* collection comprises half a gigabyte of syntactically analyzed German noun phrases [17]. Both hand-crafted and synthetic query sets were evaluated against the three corpora, resulting in four test suites. Unlike *CitiesM* with 90 manual queries on the *Cities* collection, *CitiesA* (639 queries), *XMarkA* (192 queries), and *NpA* (571 queries) consist of automatically generated queries on the *Cities, XMark,* and *NP* collections, respectively. Only path queries were considered in order to minimize dependencies on the underlying evaluation algorithm.

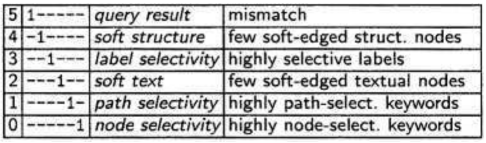
Table 1. Document collections



name	XML size	*nodes*	keywords	*label paths*	depth
Cities	1.3 MB	*16,000*	19,000	*253*	7
XMark	30 MB	*417,000*	84,000	*515*	13
NP	510 MB	*4,585,000*	130,000	*2,349*	40

For a systematic analysis of the results, all queries were classified according to six *query characteristics,* summarized in table 2 (see [21] for an extended scheme). Each characteristic of a given query is encoded by one bit in a *query signature* determining which class the query belongs to. A bit value of 1 indicates a more restrictive nature of the query w.r.t. the given characteristic, whereas 0 means the query is less selective and therefore harder to evaluate. Hand-crafted queries were classified manually, whereas synthetic queries were assigned signatures automatically during the generation process. Three groups of query characteristic turned out to be most interesting for our purposes. First, bit 5 distinguishes satisfiable (`0-----`) from unsatisfiable (`1-----`) queries (read "`-`" as "don't care"). Bits 4 to 2 concern the navigational effort during evaluation: queries with `-000--` signatures, being structurally unselective, cause many index paths to be visited. Finally, bits 1 and 0 characterize keyword selectivity, a common IR notion which we have generalized to structured documents: A keyword is called *node-selective* if there are few document nodes containing that keyword, and *path-selective* if there are few index nodes referencing such document nodes (for details on the collection-specific selectivity thresholds, see [21]). For instance, the query classes `0---10` contain satisfiable queries whose keywords occur often in the documents, though only under a small number of different label paths. All 64 path query classes are populated in the test suites, and only few with less than three queries.

Table 2. Path query classification scheme



5	1-----	*query result*	mismatch
4	-1----	*soft structure*	few soft-edged struct. nodes
3	--1---	*label selectivity*	highly selective labels
2	---1--	*soft text*	few soft-edged textual nodes
1	----1-	*path selectivity*	highly path-select. keywords
0	-----1	*node selectivity*	highly node-select. keywords

The index structures were integrated into the XML retrieval system X^2 [15]. Since large parts of the query evaluation algorithms and even index algorithms are shared by all index structures, the comparison results are not polluted with implementational artefacts. All tests have been carried out sequentially on the same computer (AMD Athlon™ XP 1800+, 1 GB RAM, running SuSE Linux 8.2 with kernel 2.4.20). The PostgreSQL relational database system, version 7.3.2, was used as relational backend for storing the index structures (with database cache disabled). To compensate for file system cache effects, each query was processed once without taking the results into account. The following three iterations of the same query were then averaged.

5.2 Results

Figure 9 shows the performance results for three selected sets of query classes, as indicated by the plot titles: while the left column covers all evaluated queries (`------`), the one in the middle narrows down to those queries with mostly unlabelled soft edges (`-000--`) The right column is restricted to all matching queries (`0-----`). The three bar charts in the upper row depict, on a logarithmic scale, the *average speedup* of ICADG (▨) and SCADG (■) over the DataGuide, i.e. the ratio of the DataGuide's to the respective CADG's evaluation time. For each of the four test suites, the speedup was averaged in three steps: first over the three iterations of each query, then over all queries in each query class, and finally over all classes selected for the given plot. This step-wise averaging ensures that query classes of different cardinality are equally weighted. The three plots in the second row depict the ICADG's *speedup distribution,* again on a logarithmic scale. For each test suite, the corresponding graph indicates for how many queries (in percent of the total number of queries in the selected classes; ordinate) the ICADG achieved a specific speedup over the DataGuide (abscissa). As indicated by the position of the symbols (+,□,◇,×), queries have been grouped into five speedup intervals ($(-\infty, 1)$; [1,2); [2,10); [10,100); $[100, \infty)$). For convenience, the distributions are given as interpolating functions rather than as histograms.

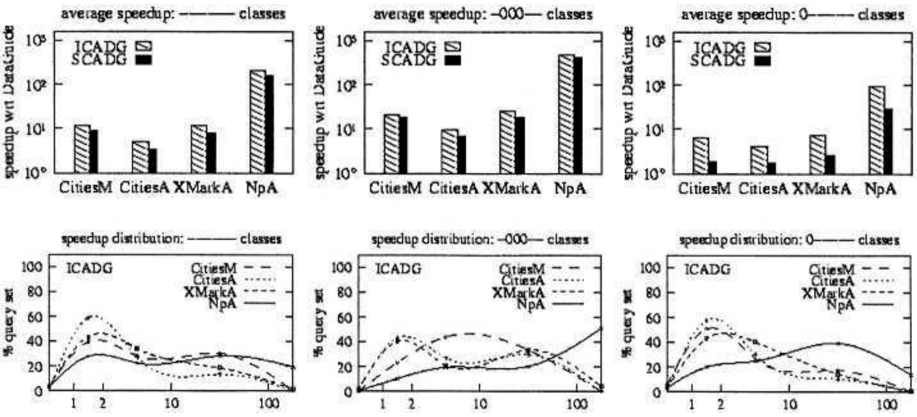


Fig. 9. Retrieval time CADG vs. DataGuide

As shown in the upper left chart in figure 9, the ICADG always performs a little better than the SCADG, beating the DataGuide by a factor 5 in the worst case *(CitiesA).* In the *CitiesM* and *XMarkA* test suites, the ICADG reaches an average speedup of one order of magnitude. On the most challenging collection, *NP*, it is 200 times faster than the DataGuide. As expected, the speedup increases for poorly structured queries (upper middle), where the potential for path pruning is higher. The speedup gain (ICADG 80-140%, SCADG 90-160%) grows with the size of the corpus. In *NpA*, the ICADG evaluates structurally unspecific queries 479 times faster than the DataGuide on average. The distribution plot for `-000--` queries (lower middle) illustrates how under these conditions the bulk of queries is shifted from [1,2) to [2,10) for *CitiesM* (— —), and to [10,100) for *CitiesA* (······) and *XMarkA* (- - - -). For *NpA* (——), the portion of [1,2) queries drops to 10% (formerly 28%), while the $[100, \infty)$ portion jumps from 19% to 51%, i.e. the ICADG processes one out of two queries by two orders of magnitude faster than the DataGuide. Higher keyword selectivity (`----11` and `-00011`, omitted) again increases the speedup by 10-20% on average, and up to 30% for the ICADG. Yet content awareness pays off even for unselective keywords.

The two plots in the right column of figure 9 focus on the subset of satisfiable queries (`0-----`). It makes up 50% of the test suites. While the ICADG's speedup still reaches 4-7 in the smaller test suites (vs. 5-12 for all queries) and two orders of magnitude in *NpA*, the SCADG performs only twice as good as the DataGuide on the *Cities* and *XMark* collections. On *NP* it beats the DataGuide by one order of magnitude (avg. speedup 28). Obviously the SCADG excels at filtering out unsatisfiable queries, especially those with non-existing keywords which it rejects in retrieval phase 0. In practice this is a valuable feature, as users are unwilling to accept long response times when there is no result in the end. Moreover, a suitable evaluation algorithm might list query hits incrementally, allowing users to browse the result while evaluation continues. Yet for unsatisfiable queries there is nothing to display, such that every second of the response time is lost.

The experiments also prove that both CADGs are most effective for large document collections such as *NP,* in terms of both retrieval time and storage. As shown in figure 10, the ICADG grows to 87% (2.4 MB) and the SCADG to 168% (4.6 MB) of the size of *Cities* in the database (DataGuide 1.6 MB). However, this storage overhead is reduced considerably for *XMark* and completely amortized for *NP* (ICADG 3% (21 MB), SCADG 6% (36 MB), DataGuide 3% (15 MB)). The SCADG's overhead over the ICADG, with 64-bit signatures in the signature table and index tree, ranged from 2 MB *(Cities)* to 16 MB *(NP).* Note that the storage measurements include so-called *function words* (i.e. extremely unselective keywords without a proper meaning) and inflected forms, which may be excluded from indexing using common IR techniques like stop word lists and stemming [9]. This further reduces the storage overhead. The resulting index, although inexact, is well suited for XML result ranking [10,18,12,22].

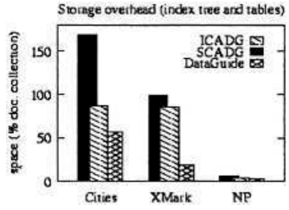


Fig. 10. Index size

6 Related Work

In this section we focus on related XML index structures with support for textual content. Work on index structures for XML in general is surveyed in [20].

Closest in spirit to the CADG approach are the *IndexFabric* [6] and the *Joined Lists* approach proposed in [12]. Based on the DataGuide, both provide a link between index nodes and keyword occurrences, a crucial concept for efficient query evaluation. In the case of the IndexFabric, this link is explicit: the index nodes are enriched with Tries representing textual content. This is equivalent to the CADGs' materialized join. Additionally, the IndexFabric is equipped with a sophisticated layered storage architecture, but provides no content-aware navigation like CADGs. Joined Lists are inverted files that are joined during query evaluation. Index nodes and keyword occurrences are linked indirectly with references from each keyword occurrence to the corresponding index node. Unlike CADGs, Joined Lists involve neither materialized joins nor content-aware navigation, relying on suitable join algorithms for efficient query evaluation.

The *BUS index* [19] integrates structured document retrieval and relevance ranking based on term frequency in document nodes. Keywords are mapped to the containing document and index nodes. This results in a content/structure join at query time, albeit at index node rather than document node level. CADGs abandon this join and use content awareness to minimize path matching and I/O.

The *Signature File Hierarchy* [5] provides content-aware path matching like the SCADG. However, without signatures for index nodes, or text blocks as proposed in [8], a new government signature must be fetched from disk for each document node during path matching. This entails a significant I/O overhead. Pruning is less effective by lack of containment signatures in the index tree.

More IR-oriented approaches like [7] tend to use the document structure for determining which parts of the data to retrieve and for ranking, but not for path matching. Label paths are therefore not represented as a navigable index tree.

The *Context Index* [16], extending inverted-file based indices, uses *structure signatures* to discard mismatches early during retrieval. They hold approximate information about the structural context (label path) of keyword occurrences.

Materialized Schema Paths [3] represent the structural context of keywords, entire text portions, and node labels statistically. In a content-centric approach (see section 3.1), they can be used to index keyword-specific corpus fragments.

7 Conclusion

Results. The Content-Aware DataGuide (CADG) is an efficient index for textual XML data. Combining structure and text matching during all retrieval phases by means of standard IR and DB techniques, it abandons joins at query time and avoids needless I/O operations. Among the two concrete realizations of the CADG, the faster and smaller ICADG is more promising than the heuristic SCADG, in accordance with results from earlier work on flat text data [24]. Based on a novel query classification scheme, experiments prove that the ICADG outperforms the DataGuide on large corpora by two orders of magnitude on average. It is most effective for queries with little structure and selective keywords, which have been shown to be most important in real-world applications. The greatest speedup (factor 636) and lowest storage overhead (3% of the original data) is achieved for a large, heterogeneous document collection of 510 MB.

Future Work. IR and DB integration for structured document retrieval has advanced recently, but is still far from complete. We plan to adapt the CADG to XML ranking models from IR [10,18,12,22] and to enhance the $\mathbf{X^2}$ retrieval system [15] with index browsing [11] and relevance feedback. A further optimized ICADG may support keyword negation in queries. One might also investigate techniques for adaptively increasing the level of content awareness based on query statistics. An ongoing project tries to amalgamate DataGuide and CADG techniques with the rather limited indexing support for XML in commercial relational database systems. We also plan to refine the query classification scheme [21] and to assess content-aware navigation and the materialized join separately.

References

1. S. Amer-Yahia and P. Case. XQuery and XPath Full-Text Use Cases. W3C Working Draft, 2003. See `http://www.w3.org/TR/xmlquery-full-text-use-cases`.
2. R. Baeza-Yates and G. Navarro. Integrating Contents and Structure in Text Retrieval. *SIGMOD Record,* 25(1):67–79, 1996.
3. M. Barg and R. K. Wong. A Fast and Versatile Path Index for Querying Semi-Structured Data. *Proc. 8th Int. Conf. on DBS for Advanced Applications,* 2003.
4. S. Buxton and M. Rys. XQuery and XPath Full-Text Requirements. W3C Working Draft, 2003. See `http://www.w3.org/TR/xquery-full-text-requirements`.
5. Y. Chen and K. Aberer. Combining Pat-Trees and Signature Files for Query Eval. in Document DBs. *Proc. 10th Int. Conf. on DB & Expert Systems Applic.,* 1999.
6. B. Cooper, N. Sample, M. J. Franklin, G. R. Hjaltason, and M. Shadmon. A Fast Index for Semistructured Data. *Proc. 27th Int. Conf. on Very Large DB,* 2001.
7. H. Cui, J.-R. Wen, and T.-S. Chua. Hier. Indexing and Flexible Element Retrieval for Struct. Document. *Proc. 25th Europ. Conf. on IR Research,* pages 73–87, 2003.
8. C. Faloutsos. Signature Files: Design and Performance Comparison of Some Signature Extraction Methods. *Proc. ACM-SIGIR Int. Conf. on Research and Development in IR,* pages 63–82, 1985.
9. W. B. Frakes, editor. *IR. Data Structures and Algorithms.* Prentice Hall, 1992.
10. N. Fuhr and K. Großjohann. XIRQL: A Query Language for IR in XML Documents. *Research and Development in IR,* pages 172–180, 2001.
11. R. Goldman and J. Widom. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. *Proc. 23rd Int. Conf. on Very Large DB,* 1997.
12. R. Kaushik, R. Krishnamurthy, J. F. Naughton, and R. Ramakrishnan. On the Integration of Structure Indexes and Inverted Lists. *Proc. 20th Int. Conf. on Data Engineering,* 2004. To appear.
13. Q. Li and B. Moon. Indexing and Querying XML Data for Regular Path Expressions. *Proc. 27th Int. Conf. on Very Large DB,* pages 361–370, 2001.
14. J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J. Widom. Lore: A DB Management System for Semistructured Data. *SIGMOD Rec.,* 26(3):54–66, 1997.
15. H. Meuss, K. Schulz, and F. Bry. Visual Querying and Explor. of Large Answers in XML DBs with X^2. *Proc. 19th Int. Conf. on DB Engin.,* pages 777–779, 2003.
16. H. Meuss and C. Strohmaier. Improving Index Structures for Structured Document Retrieval. *Proc. 21st Ann. Colloquium on IR Research,* 1999.
17. J. Oesterle and P. Maier-Meyer. The GNoP (German Noun Phrase) Treebank. *Proc. 1st Int. Conf. on Language Resources and Evaluation,* 1998.

18. T. Schlieder and H. Meuss. Querying and Ranking XML Documents. *JASIS Spec. Top. XML/IR 53(6):489-503,* 2002.
19. D. Shin, H. Jang, and H. Jin. BUS: An Effective Indexing and Retrieval Scheme in Structured Documents. *Proc. 3rd ACM Int. Conf. on Digital Libraries,* 1998.
20. F. Weigel. A Survey of Indexing Techniques for Semistructured Documents. Technical report, Dept. of Computer Science, University of Munich, Germany, 2002.
21. F. Weigel. Content-Aware DataGuides for Indexing Semi-Structured Data. Master's thesis, Dept. of Computer Science, University of Munich, Germany, 2003.
22. J. E. Wolff, H. Flörke, and A. B. Cremers. Searching and Browsing Collections of Structural Information. *Advances in Digital Libraries,* pages 141–150, 2000.
23. XML Benchmark Project. A benchmark suite for evaluating XML repositories. See `http://monetdb.cwi.nl/xml`.
24. J. Zobel, A. Moffat, and K. Ramamohanarao. Inverted Files Versus Signature Files for Text Indexing. *ACM Transactions on DB Systems,* 23(4):453–490, 1998.

Performance Analysis of Distributed Architectures to Index One Terabyte of Text

Fidel Cacheda[1], Vassilis Plachouras[2], and Iadh Ounis[2]

[1] Departament of Information and Communication Technologies, University of A Coruña
Facultad de Informática, Campus de Elviña s/n, 15071 A Coruña, Spain
fidel@udc.es
[2] Department of Computing Science, University of Glasgow
Glasgow, G128QQ, UK
{vassilis, ounis}@dcs.gla.ac.uk

Abstract. We simulate different architectures of a distributed Information Retrieval system on a very large Web collection, in order to work out the optimal setting for a particular set of resources. We analyse the effectiveness of a distributed, replicated and clustered architecture using a variable number of workstations. A collection of approximately 94 million documents and 1 terabyte of text is used to test the performance of the different architectures. We show that in a purely distributed architecture, the brokers become the bottleneck due to the high number of local answer sets to be sorted. In a replicated system, the network is the bottleneck due to the high number of query servers and the continuous data interchange with the brokers. Finally, we demonstrate that a clustered system will outperform a replicated system if a large number of query servers is used, mainly due to the reduction of the network load.

1 Introduction

Retrieval systems based on a single centralized index are subject to several limitations: lack of scalability, server overloading and failures [6]. Therefore, given these facts, it seems more appropriate to turn to the distributed Information Retrieval (IR) systems approach for the storage and search processing.

In a distributed search environment, there are usually two basic strategies for distributing the inverted index over a collection of query servers. One strategy is to partition the document collection so that each query server is responsible for a disjoint subset of documents in the collection (called local inverted files in [13]). The other option is to partition based on the index terms so that each query server stores inverted lists corresponding to only a subset of the index terms in the collection (called global inverted files in [13]). The study in [15] indicates that the local inverted file organization uses system resources effectively, provides good query throughput and is more resilient to failures.

From the database point of view, a distributed information retrieval system could follow a single database model or a multi-database model [4]. In the single database model, the documents are copied to a centralized database, where they are indexed and made searchable. In a multi-database model, the existence of multiple text

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 394–408, 2004.

databases is considered explicitly, and at the same time, each database could have the inverted index distributed.

This work is a case study of different architectures for a distributed information retrieval system on a very large Web collection. The SPIRIT collection (94,552,870 documents and 1 terabyte (TB) of text) [10] is used for the simulation of the distributed IR system. We partition the collection of documents using a local inverted file strategy, and we test the response times for different configurations. Although the timings obtained depend on the specific system simulated [1], the trends and conclusions should be independent of the system used. In this way, our study works out the required resources and the best architecture to handle a very large collection of data, like SPIRIT. We believe that this work is a step along the recent trend in building very large collections for Web IR, like the TREC Terabyte track[1] initiative.

The improvements in the performance of a single database model are examined in a distributed and replicated system. Also, the effects of a multi-database model are tested through a clustered system.

We start by presenting the related work. In Section 3, we describe the simulation models used (analytical, collection and distributed models). Next, we describe the simulations performed for the different architectures: distributed, replicated and clustered system, and the results obtained. Finally, the main conclusions are presented.

2 Related Work

The work on architecture performance is the most directly related to this paper. Several articles [2], [5], [12] analyze the performance of a distributed IR system using collections of different sizes and different system architectures. Cahoon and McKinley in [3] describe the result of simulated experiments on the distributed INQUERY architecture. Using the observed behaviour for a mono-server implementation, they derive the performance figures for a distributed implementation, proving it to be scalable.

The previous work for distributing the inverted index over a collection of servers is focused on the local and global inverted files strategies [13], [15], showing that the local inverted file is a more balanced strategy and a good query throughput could be achieved in most cases.

Our work is focused on the performance evaluation of several distributed architectures using a massive cluster of workstations (up to 4096) and identifying the limitations of each model. The novelty of this work relies on the size of the collection represented (1TB of text) and the large number of workstations simulated. This work is especially related to [3] and [13], but it differs mainly in three points. First, the probabilistic model is considered and therefore, disjunctive queries are used in the system (without the reduction in the answer set provided by the conjunctive operations). Second, a simple analytical model is developed initially for a single-database/single-server environment (similarly to [13]), and this will be the basis for the simulation of the distributed IR systems, composed of multiple query servers (similarly to [3]). Third, initially the results of the analytical model are tested using

[1] http://www-nlpir.nist.gov/projects/terabyte/

the TREC WT10g collection and the set of the topic relevance queries from TREC10 [7]. Next, a document collection of 1TB and its queries are modelled in order to obtain more generic results.

3 Simulation Model

To explore the performance of different architectures for a distributed IR system, we implemented a discrete event-oriented simulator using the JavaSim simulation environment [11].

The simulation model defined in this work is divided into three parts. Initially an analytical model has been developed for the simulation of a simple IR system based on the WT10g collection and its set of real queries using a single server. Next, a collection model is defined to simulate, in general, the behaviour of any collection of documents and in particular, a new collection composed of 94 million documents and 1TB of text. Finally, the basic IR model is extended to a distributed IR model defining the behaviour of a local area network of computers and modelling the tasks of the query brokers and the query servers.

3.1 Analytical Model

In this section, we describe a simplified analytical model for the querying process in the IR system described in [1], using the WT10g collection and the set of queries used for TREC10 [7]. This analytical model is similar to the one described in [13].

A series of experiments were carried out to identify and estimate the basic variables and critical parameters of the analytical model. The notation for these variables and parameters is provided next:

q_i: vector of keywords for the i*th* query.
k_i: number of keywords in query q_i.
d_k: number of documents of the inverted list for keyword *k*.
r_i: number of results obtained in query q_i.
tc_1: first coefficient for the time to compare two identifiers and swap them.
tc_2: second coefficient for the time to compare two identifiers and swap them.
ti: initialisation time, including memory allocation and output display.
ts: average seek time for a single disk.
tr: average time to read the information about one document in an inverted list and do its processing (seek time is excluded).
t_i: total time (in milliseconds) to complete processing of the query q_i.

Once the query server receives the query vector q_i for processing, it reads from disk the inverted lists associated with the k_i keywords, whose length is given by d_k. Then the inverted lists are merged and sorted to form the answer set whose length is given by r_i. Previous works [13] have modelled this using a linear relationship, but from our experiments, a logarithmic model seems to fit more accurately as the number of results increases (coefficients tc_1 and tc_2). Hence, the time to merge and sort *n* results

(tc) is calculated as: $tc = tc_1 \times n + tc_2 \times \ln(n)$ (versus the linear model used in [13] $tc = tc_1 \times n$).

The processing of a query is divided into four phases: an initialization phase, seeking disks, reading the inverted lists from disk and assigning weights to documents, and obtaining the answer and ranking the results. Therefore, the processing time for a query q_i is t_i, given by:

$$t_i = ti + k_i \times ts + \sum_{k \in q_i} d_k \times tr + tc \times r_i .$$

In this analytical model, the parameters d_k and r_i have to be estimated accurately. We evaluate the accuracy of the estimation by processing the topic relevance queries from TREC10's Web track with a real IR system [1] in order to obtain the exact number of results for each query and the exact number of documents associated with each of the inverted lists retrieved.

The accuracy of the analytical model was confirmed comparing the response times of the real IR system for the queries number 501 to 550 from the WT1Og collection (the parameter values used are: ti=1400ms, ts=0.03ms, tr=4.0208µs, tc_1=0.000131, tc_2=0.000096; q_i, k_i, d_k, r_i depend on the collection modelled or simulated, as described in the next section).

3.2 The Spirit Collection Model

The basic analytical model defined for the WT10g collection will be extended to work with synthetic databases and queries. The objective is to simulate the so-called SPIRIT collection, composed of approximately 94 million Web documents and 1TB of text, although no queries and relevant assessments exist for the moment [10]. We divide this collection in 72 sub-collections and we use the statistical information (vocabulary size, document size, etc.) of one of them to model the whole collection.

Document Model

For the document model, we first study the main parameters of one of the sub-collections, and using this as a basis, the values for the whole SPIRIT collection are estimated. In Table 1, we provide the definition for the parameters considered.

Table 1. Parameters for the document model. The real values were obtained from a SPIRIT subcollection and the estimated values represent the whole SPIRIT collection

Parameter	Real values	Estimated values	Description
D	1,221,034	94,552,870	The number of documents
W	456	456	Average words per document
T	4,301,776	73,689,638	Total words in V, i.e. $T = \|V\|$
$F(w)$	$Z_1(w)$	$Z_2(w)$	Pr(word = w)

The first column describes the parameters that represent a database of documents. The database consists of a collection of *D* documents. Each document is generated by a sequence of *W* independent and identically distributed words. *V* represents the vocabulary, where each word is uniquely identified by an integer *w* in the range $1 \leq w \leq T$, where $T = |V|$. The probability distribution *F* describes the probability that any word appears and, for convenience, is arranged in decreasing order of probability.

The second column of the table represents our base case scenario and the values are obtained from one fraction of the SPIRIT collection. To define a specific probability distribution Z_l of *F,* a distribution is fitted to the rank/occurrence plot of the vocabulary, and then normalized to a probability distribution. The regression analysis performed confirms that the quadratic model fits better the real distribution *(R = 0.99770)* versus the linear model representing the Zipf's law *(R = 0.98122).* The quadratic model is similar to Zipf's, although in previous works [15], it has proved to match the actual distribution better. Given the quadratic fit curve, the form of the probability distribution $Z_l(w)$ is obtained from the quadratic model, divided by a normalisation constant [15].

The third column of Table 1 shows the values for the parameters of the whole SPIRIT collection. The number of documents in the collection is 94,552,870. The average number of words per document is supposed to remain stable. Therefore, the same value as the base case was chosen.

The size of the vocabulary of a collection of documents matches the Heaps law [8], with $K = 4.60363$ and $\beta = 0.6776$. Therefore, an approximation of 73,689,638 unique terms for the whole collection is obtained.

Finally, a different probability distribution is provided for the whole collection. Given the quadratic fit curve previously described, a new normalization constant is defined for the new vocabulary size ($\Theta = 4.294476 \times 10^8$).

Query Model

A query is a sequence of terms $(t_1,\ldots,t_k)$ generated from *K* independent and identically distributed trials from the probability distribution *Q(t).* Actually, in our simulation study the number of terms is selected uniformly between 1 and 4 terms per query. In Table 2 a description of each parameter and the base values chosen are presented.

The most realistic query model is the skewed query model [9], where Q is modelled assuming that the probability of a term occurring in a query is proportional to that term's frequency in vocabulary. The probability distribution *Q(t)* for the skewed query models is: (where *C* represents a normalization constant)

$$Q(t) = \begin{cases} C \times Z(t) & \textit{if } sT \leq t \leq (u-s)T \\ 0 & \textit{otherwise} \end{cases} \quad \textit{where } 1 = \sum_{i=sT}^{(u-s)T} C \times Z(i) \textit{ and } u \geq 2s\,.$$

The parameters *u* and *s* affect the probability that a term appears in a query. As *u* decreases, the probability of choosing a word of high rank increases. Words of high rank are highly repeated in the documents of the collection. Therefore, if *u* is too small, the queries will retrieve a large fraction of the documents. On the other hand, if *u* is too large, the answer sets will be too small [9]. The parameter *s* is introduced to avoid the effect of the first words in the rank, i.e. stopwords, which increase excessively the number of results obtained. As *s* increases more words from the top rank are considered to be stopwords, and therefore are not used as query terms. In all our simulations the values for these parameters are: $u = 0.01$ and $s = 0.00007$.

Table 2. Parameters for the query model

Parameter	Value	Description
K	[1-4]	The number of terms per query
$F_q(t)$	$Q(t)$	Pr(term = t)
u		The fraction of T used in the query terms
s		The fraction of T skipped for.the query terms

At certain points in the simulation, we will need to know the expected size of an inverted list and the expected size of an answer set. Let us assume a query with terms $t_1,\ldots, t_k$ that is executed in a collection (or sub-collection) of documents of size *Documents*. If we are considering the whole collection *Documents* = *D*, but in a distributed environment, *Documents* corresponds to the number of documents covered by each of the distributed indices. So, the number of documents of an inverted list for term t_i will be [15]: $\mathit{Documents} \times [1-(1-Z(t_i))^w]$.

Consequently, the expected size of the answer set for a query with terms $t_1,\ldots,t_k$ (supposing a disjunctive query) is:

$$\mathit{Documents} \times [1-(1-Z(t_1))^w \times \ldots \times (1-Z(t_k))^w].$$

In order to test the accuracy of the described SPIRIT document collection, a simulation was performed to replicate the results for the WT10g collection, with the analytical model. The results showed that the simulations using the skewed query model produce on average similar response times as the real queries. Although the fluctuations of the real queries are not present in the skewed query model, this model can be considered quite accurate for the average response times.

3.3 Distributed Model

In a distributed IR system, the queries are stored on a global queue, which is controlled by one or more *central brokers*. Each broker will take one query and will send it to all the query servers through a network, in a local index organization [13]. Each query server then processes the whole query locally, obtains the answer set for that query, ranks the documents, selects a certain number of documents from the top of the ranking and returns them to the broker. The broker collects all the local answer sets and combines them into a global and final ranked set of documents.

We assume that the system is operating in batch mode and that there are always enough queries to fill a minimum size query processing queue (by default, 50 queries).

The analytical model previously described is now extended to support a distributed IR system, with local index organization. Some new parameters are defined:

Table 3. Parameters for the distributed model

Parameter	Value	Description
LANOverhead	0.1ms	Network overhead for each packet sent
LANBandwidth	100Mbps	Network speed (in bits per second)
QuerySize	100 bytes	Number of bytes sent from the broker to the query servers for each query request
DocAnswerSetSize	8 bytes	Number of bytes per document sent in the local answer sets (document id and document ranking)

$d_{k,j}$: number of documents of the inverted list for keyword k on query server j.

$r_{i,j}$: number of results obtained for query q_i on query server j.

tr_{max}: maximum number of top ranked documents returned as the local answer set (we consider the top 1000 documents only).

$tr_{i,j}$: number of documents from the top ranking in query q_i returned as the local answer set for query server j.

$t_{i,j}$: total time to complete the processing of query q_i at query server j.

$rq_{i,j}$: time to receive the query q_i for the query server j.

$ra_{i,j}$: time to receive the local answer set for query q_i from the query server j.

Therefore, the time for the query server j to process the query q_i is given by:

$$t_{i,j} = rq_{i,j} + ti + k_i \times ts + \sum_{k \in q_i} d_{k,j} \times tr + tc \times r_{i,j} .$$

The parameters $d_{k,j}$ and $r_{i,j}$ are estimated through the collection model parameters d_k and r_i respectively, described in the previous section.

As soon as the broker has received all the local results from all the query servers, it must combine them to obtain the final answer set. Therefore, the total processing time for query q_i could be given by:

$$t_i = \max(t_{i,j}) + \max(ra_{i,j}) + \sum_j tr_{i,j} \times tc .$$

The problem is that the parameters $rq_{i,j}$ and $ra_{i,j}$ can not be estimated using an analytical model as they depend directly on the network load of each moment. Therefore, it is necessary to capture the behaviour of the network to represent accurately the response times of a distributed IR system.

In our case, the system will contain a single LAN that will be simulated by a single FCFS infinite length queue. This LAN will manage all the messages sent by the brokers to the query servers and the answers from the query servers to the brokers. The service time for a request is calculated by the equation:

$$LANOverhead + RequestLength \times \left(LANBandwitdh / 8\right)^{-1} \times 1000 .$$

The values and description for the parameters used in the simulation of the network are described in Table 3. The *RequestLength* parameter depends on the type of message sent. If a query is sent to the query servers, the value of the *QuerySize* parameter will be used. If the local answer for query q_i set is sent from query server j to the broker, then the length of the packet will be: $tr_{i,j} \times DocAnswerSetSize$.

4 Simulation Results

This section describes the results of several experiments developed using the simulation model described in the previous section. The objective is to determine different approaches for the distribution and replication of the collection using a bunch of query servers and compare the performance between the different configurations.

All the simulations are based on the 1TB SPIRIT collection model [10]. The queries have been modelled using the skewed query model and following a worst case scenario: each query will retrieve on average approximately 8.4 million documents (9% of the whole collection). A batch of 50 queries is used to test the performance, and for each different configuration, 5 different simulations (with distinct initial seeds) are run, and the average values for the execution times are calculated for each query.

Initially, a purely distributed system is examined. Next the effects of the replications are analyzed and then, we examine possible configurations of a clustered system (based on an asymmetric distribution and replications).

4.1 Distributed System

In this set of experiments, the collection of documents is distributed using the local index organization over *N* query servers, where *N = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 768 and 1024*. Initially, the number of brokers for the distributed IR system is set to one, and next is increased to 2, 3 and 4. The results are displayed in Fig. 1.

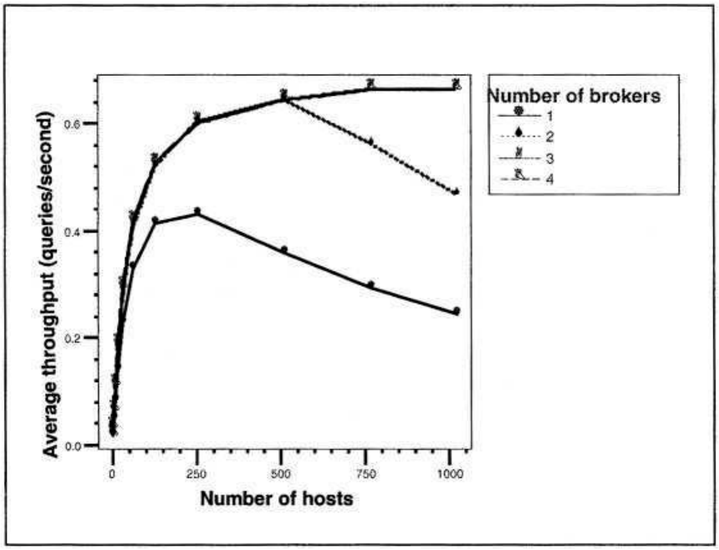
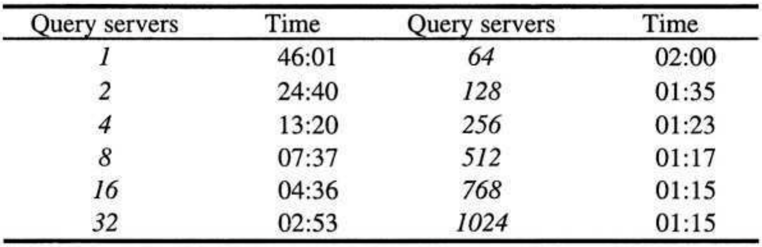


Fig. 1. Throughput for the simulation of a distributed IR system with local index organization

Table 4. Estimated time (mm:ss) to process 50 queries by the distributed system (3 brokers)



Query servers	Time	Query servers	Time
1	46:01	*64*	02:00
2	24:40	*128*	01:35
4	13:20	*256*	01:23
8	07:37	*512*	01:17
16	04:36	*768*	01:15
32	02:53	*1024*	01:15

The optimal performance is achieved when two or more brokers are used (see Fig. 1). In fact, with less than 512 query servers, two brokers are able to provide continuously queries to the servers, and so, the performance of the system is maximised. However, there is still a bottleneck with 768 or 1024 query servers, with inactivity periods that will reduce the throughput. Three brokers will provide the maximum throughput, and no benefit is obtained if the number of brokers is increased.

The bottleneck in the brokers is due to the number of local answer sets received from all the query servers that must be sorted. Therefore, increasing the number of query servers will benefit the processing time in the query servers as each server reduces the size of its index. On the other hand, the brokers will receive more local answer sets to be merged in the final result set. In fact, if the number of query servers is high enough, the performance will start descending at a certain point, independently of the number of brokers used.

Working with an optimal configuration of three brokers, Table 4 provides an estimation of the expected time in minutes to process 50 queries by a distributed IR system, using from 1 to 1024 query servers.

Without any improvements, the throughput tends to be stabilised around 0.64 queries/second with 512 query servers, with minor improvements as the number of servers increases (0.66 queries/second with 1024 query servers).

4.2 Replicated System

A replicated system is composed of one or more distributed IR systems. Each distributed system indexes the whole collection, and all the distributed systems replicated have the same number of query servers. In this case, the distributed system previously described could be seen as a replicated system, with only one replica.

In a replicated system, the brokers must decide initially which replica will process the query, and then broadcast the query to all the query servers in the replica. The objective of the selection of the replicas is to balance the load through all the replicas to obtain an optimal performance for the whole system. In our case, a round robin policy is used to distribute the queries to the replicas. Each broker will select a different initial replica and for each following query the next replica is selected.

Firstly, the optimal number of brokers required in a generic replicated system is analysed. To study this, a set of replicated systems was simulated, changing the number of brokers used. A summary of the results is provided in Table 5.

Table 5. Throughput (queries per second) for different replicated IR systems. The "Query servers" column represents the number of servers per replica. Each column indicates the number of replications (R) and the number of brokers used (B)

Query	R=1	R=2			R=3			R=4		
servers	B=3	B=4	B=5	B=6	B=6	B=7	B=8	B=8	B=9	B=10
1	**0.02**	0.03	**0.03**	0.03	0.05	**0.05**	0.05	0.06	**0.06**	0.05
2	**0.03**	0.05	**0.06**	0.06	0.08	**0.09**	0.08	0.11	**0.11**	0.11
4	**0.06**	0.1	**0.11**	0.11	0.14	**0.15**	0.16	0.19	**0.19**	0.2
8	**0.11**	0.18	**0.2**	0.2	0.27	**0.27**	0.29	0.35	**0.36**	0.36
16	**0.18**	0.3	**0.34**	0.35	0.47	**0.47**	0.52	0.61	**0.64**	0.63
32	**0.29**	0.5	**0.53**	0.55	0.73	**0.77**	0.8	0.98	**0.99**	0.99
64	**0.41**	0.75	**0.78**	0.81	1.1	**1.18**	1.1	1.46	**1.48**	1.47
128	**0.52**	1	**0.98**	1	1.46	**1.42**	1.4	1.95	**1.93**	1.9
256	**0.6**	1.17	**1.16**	1.18	1.72	**1.7**	1.73	2.2	**2.18**	2.26
512	**0.64**	1.19	**1.24**	1.27	1.6	**1.78**	1.81	1.95	**2.05**	2.13
768	**0.66**	0.96	**1.24**	1.29	1.17	**1.42**	1.46	1.26	**1.45**	1.47
1024	**0.66**	0.85	**0.99**	1.07	1.06	**1.08**	1.11	1.11	**1.13**	1.13

Initially, a two replications system is simulated, testing different number of brokers. With only four brokers, there is a reduction in the performance, following the pattern of the basic distributed system with two brokers (decreasing with 768 or 1024 hosts per replica). While, with five brokers the maximum throughput is achieved, an increase in the number of brokers will slightly increase the performance (and simultaneously, the network load).

The case of the systems with three and four replications is quite similar. With six and eight brokers, there is a decrease in the performance for more than 512 hosts, reproducing the behaviour of one unique distributed system. As in the previous case, one more broker is sufficient to avoid the bottleneck and serve properly all the servers.

Generally, for the configurations simulated, the number of brokers necessary to operate a generic replicated system, with R replicas, is given by: $2R + 1$. With $2R$ brokers there is still a bottleneck when the number of query servers is high, and this extra broker will reduce the idle times in the hosts. If the number of replications is further increased, more extra brokers would be necessary to maintain throughput at the same levels.

Another important point in the replicated systems is the relation between the throughput and the number of replicas. If a basic distributed system has a throughput of T queries/minute, then the expected throughput for a system with R replicas will be $T*R$. This is coherent with the results obtained in Table 5, especially considering 128 or less query servers per replica. In this case, the throughput obtained for the different replicated systems, with the optimal number of brokers (or more), is slightly below the theoretical value. This is due to the round robin distribution policy used in the brokers, as it can lead to some small periods of inactivity at certain replicas. In future works, some other distribution policies can be analysed in order to improve the throughput up to the optimal theoretical value, similar to the one used in [12].

Note that if more than 256 query servers are used per replica, the performance of the system starts to decrease rapidly. If the total number of query servers in the system (considering all the replicas) is beneath 1000, the performance is improved with each new replica added. However, if the number of query servers is over this limit, the performance decreases, especially as more replicas are included in the system. In fact, a system with 4 replicas of 1024 query servers has a worse throughput than a system with 4 replicas of 64 servers each.

This loss of performance is due to the network. Each replication adds more hosts to the network, which is used intensively to send the results back to the brokers. As a consequence, the network latency is greatly increased with each new replica added, converting the network to the bottleneck for the whole system. In a system with one replica and 1024 query servers, each byte will reach its destination in 0.36 ms on average. However, in a system with four replicas and 1024 query servers per replica, the time each byte needs to reach its destination increases 10 times. Hence, all the messages sent through the network are highly delayed producing inactivity periods on both, the query servers and the brokers.

4.3. Clustered System

A clustered system is divided into groups of computers, where each group operates as an autonomous distributed and replicated IR system. Each cluster can be composed of a different number of query servers. We assume that each cluster is responsible for one disjoint part of the whole collection of documents, and each cluster could use distribution and replication to store its respective index.

The brokers are global for the whole IR system. First, a broker must determine the appropriate cluster for each query and then should broadcast the query to the selected clustered system. If the cluster supports replication, then the broker will also decide to which replica the query will be sent (e.g. by using the round robin policy).

Different commercial Web IR systems claim to use a clustered system adapted to the distributions of the queries received (e.g. AllTheWeb). Therefore, the objective of these experiments is to test if the performance of a replicated IR system could be improved using a clustered system fitted to a distribution of queries, and how the changes of this distribution will affect the performance.

In the work by Spink et al. [14], a set of real queries of Web users is categorized into 11 different topics. Moreover, the variations in the percentage of queries for each topic are analyzed in three different years: 2001, 1999 and 1997. Table 6 provides a summary of the 11 topics and the percentage of queries through the different years.

In the simulated systems, once a query is generated, it is automatically assigned to a topic using these distributions values. In these simulations, the number of queries is increased to 200 in order to examine the whole range of topics, and the queries will retrieve 3 million documents on average to fit the size of the clusters.

In these experiments, we assume that each topic is indexed in a different cluster. The collection is divided into 11 sub-collections with an inverted file of approximately the same size, that is 8.5 million documents and, therefore, the 11 clusters defined will index the same number of documents, although using a different number of servers.

Table 6. Distribution of queries across general topic categories, and configurations for the clustered systems simulated

Topics	2001	1999	1997	Config 1	Config 2
Commerce	24.755 %	24.73 %	13.03 %	8 * 4	63 * 4
People	19.754 %	20.53 %	6.43 %	6 * 4	51 * 4
Non-English	11.355 %	7.03 %	3.84 %	5 * 3	39 * 3
Computers	9.654 %	11.13 %	12.24 %	4 * 3	33 * 3
Pornography	8.555 %	7.73 %	16.54 %	5 * 2	44 * 2
Sciences	7.554 %	8.02 %	9.24 %	5 * 2	38 * 2
Entertainment	6.655 %	7.73 %	19.64 %	4 * 2	34 * 2
Education	4.554 %	5.52 %	5.33 %	6 * 1	47 * 1
Society	3.955 %	4.43 %	5.44 %	5 * 1	41 * 1
Goverment	2.054 %	1.82 %	3.13 %	3 * 1	21 * 1
Arts	1.155 %	1.33 %	5.14 %	2 * 1	12 * 1

The base sub-collection of 8.5 million documents has been distributed over *N* query servers, where *N = 1, 2, 4, 8, 16, 32, 64, 128, 256 and 512.* The throughput matches the previous results displayed in Fig. 1, with an optimal configuration of two brokers.

Roughly speaking, two different configurations have been tested for the clustered system. The first one has 128 query servers and the second one has 1024 query servers. Each cluster is assigned a number of query servers proportional to the percentage of queries that it should receive (see Table 6).

For the first configuration, the baseline is a replicated IR system, with 4 replications of 32 query servers each. On the other side, the clustered system is configured in accordance with the distribution of the topics on the year 2001. In Table 6, the column *"Config 1"* describes the query servers assigned to each topic. The first number represents the number of the distributed query servers, and the second, the number of replicas in each cluster.

Figure 2 presents the box diagram for the response time for the first 100 queries processed by the tested systems. All the systems were tested for queries following the topics of years 2001, 1999 and 1997. Obviously, the performance of a replicated IR system does not depend on the type of queries (the baseline is independent of this factor), and the response times for the clustered system with 128 servers are labelled *"Config 1-2001", "Config 1-1999"* and *"Config 1-1997",* respectively.

The first clear conclusion is that the clustered system does not outperform a replicated system. The replicated system will process one query on 4682 milliseconds, while the clustered system optimally configured for the 2001 queries will just process one query in 7806 milliseconds (approximately the same performance as a system with two replicas of 64 servers). On the contrary, the clustered system reduces greatly the network load with 0.0008 ms/byte, vs. the replicated system with 0.0044 ms/byte.

On the other hand, the clustered system seems sensitive to the changes in the topics of the queries through the time. For the queries of the year 1999, the performance is nearly the same, 8068 milliseconds per query, while for the 1997, queries the performance drops to 9212 milliseconds per query. In fact, the higher differences for the topic distributions are between the years 2001 and 1997 (see Table 6).

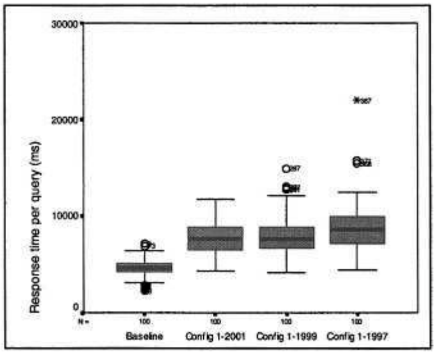


Fig. 2. Clustered IR system vs. a replicated IR system. Configuration 1: 128 query servers

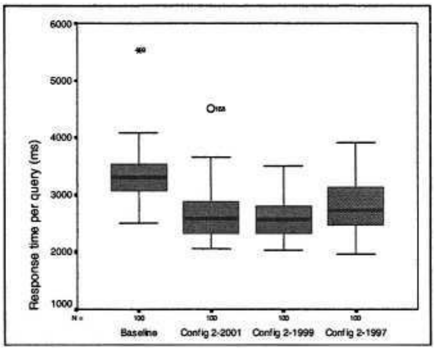


Fig. 3. Clustered IR system vs. a replicated IR system. Configuration 2: 1024 query servers

Also, the presence of atypical data reflects the effect of the changes in the topics through the time. In fact, the baseline also happens to have some atypical data but very near the confidence interval, due to the distribution of the queries over many servers. In a clustered system, with a reduced amount of hosts per cluster, some heavy queries will produce higher response times. This is more notable when the topics of the queries are changed (1997 queries), because the smaller clusters may have to cope with a higher number of queries than initially expected.

In the second configuration, the 1024 query servers of the clustered system are assigned according to the values reflected in Table 6, on the column labelled "*Config 2*". The baseline in this case, is a replicated system with 4 replicas of 256 query servers each. As for the previous case, the clustered system and the replicated system have processed the queries matching the 2001, 1999 and 1997 distributions. Fig. 3 shows the box diagram for the response time for the first 100 queries processed for all these systems.

In this case, the clustered system outperforms the baseline, for all the years of the query distributions. The replicated system has a performance of 3313 milliseconds per query, while the clustered system for the 2001 queries will process one query in 2665 milliseconds on average. Regarding the network load, while the replicated system needs, on average, 0.112 milliseconds to send one byte, the clustered system uses only 0.007 milliseconds per byte, on average.

This configuration is also sensitive to the changes in the topics of the queries, but to a smaller degree. For the 1999 queries, the performance is slightly better, 2630 milliseconds per query, but with the 1997 queries the performance drops to 2938 milliseconds per query (still outperforming the baseline).

In this configuration, the increase in the number of query servers augments the distribution of the local indexes and therefore, the increase in the response times is less significant. At the same time, the different clusters can support more easily the changes in the query topics through the time. In this configuration, for the 1997 queries, the performance decreases by 9%, while with 128 query servers the throughput decreased by 14%.

5 Conclusions

In this paper, we have described different architectures for a distributed IR system, analyzing the optimal design and estimating the maximum performance achieved with multiple configurations (from 1 up to 4096 query servers). We have studied the performance of a simulated distributed, replicated and clustered system on a very large Web collection, and we have established the bottlenecks and limitations of each possible configuration.

Two main bottlenecks have been identified in a distributed and replicated IR system: the brokers and the network. The load on the brokers is mainly due to the number of local answer sets to be sorted (characteristic of a distributed system). Therefore, the load can be improved by reducing the number of documents included in the local answer sets by all the query servers, which can affect the precision and recall parameters. Another way is to reduce the number of local lists sent to the brokers, by designing more complex and elaborate distributed protocols.

The network bottleneck is due to the high number of query servers and the continuous data interchange with the brokers, especially in a replicated IR system. The traffic over the network can be limited by reducing the number of results in each local answer set (with the additional benefit over the brokers) or compressing the local answer set before sending it.

The analysis of the clustered systems indicates that the best throughput of these systems is achieved when a great number of query servers is used, outperforming a replicated system. A clustered system will reduce greatly the network load as only a fraction of the query servers will process and answer each query. Therefore, in a replicated system, the network load increases (and the throughput improvements are slowed) as the number of servers increases. While in a clustered system the processing times in the clustered query servers could be slightly higher, the local answers will reach faster the broker and the brokers will receive fewer answers, processing the final results more efficiently.

However, the clustered systems must be configured a-priori based on the distribution of the queries that the IR system will receive. Therefore, to avoid negative effects on the performance, it is important to detect changes in the distribution of the queries through the time and re-configure the clusters of the system accordingly.

In the future, we plan to study different solutions for the brokers and network bottlenecks (e.g. distributing the brokers) and their implications in the retrieval performance. Also, these results will be used to extend the basic actual IR system to a distributed system, and in general, we believe that the results in this paper are useful to any group interested in indexing a very large collection like SPIRIT.

Acknowledgements. The work of the first author has been partially supported by Spanish CICYT, under project TIC2001-0547 and by the Fundación Caixa Galicia (grant 2002/2003 postgraduate studies).

The work of the second and third authors is funded by a UK Engineering and Physical Sciences Research Council (EPSRC) project grant, number GR/R90543/01. The project funds the development of the Terrier Information Retrieval framework (url: http://ir.dcs.gla.ac.uk/terrier).

We would also like to thank Mark Sanderson and Hideo Joho for giving us access to the 1TB dataset used for the SPIRIT Project.

References

1. Amati, G., Carpineto, C., Romano, G.: FUB at TREC-10 Web track: A probabilistic framework for topic relevance term weighting. In NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC-2001). 2001.
2. Burkowski, F. J.: Retrieval performance of a distributed database utilizing a parallel process document server. In Proceedings of the International Symposium on Databases in Parallel and Distributed Systems, pp: 71-70. 1990.
3. Cahoon, B., McKinley, K.S.: Performance evaluation of a distributed architecture for information retrieval. In Proceedings of ACM-SIGIR International Conference on Research and Development in Information Retrieval, pp: 110-118. 1996.
4. Callan, J.: Distributed information retrieval. In W. Bruce Croft, editor, Advances in Information Retrieval: Recent Research from the CIIR, chapter 5, pp: 127-150. Kluwer Academic Publishers, 2000.
5. Hawking, D.: Scalable text retrieval for large digital libraries. In Proceedings of the 1st European Conference on Research and Advanced Technology for Digital Libraries, Springer LNCS, vol. 1324, pp: 127-146. 1997.
6. Hawking, D., Thistlewaite, P.: Methods for Information Server Selection. ACM Transactions on Information Systems, 17(1), pp: 40-76. 1999.
7. Hawking, D., Craswell, N.: Overview of the TREC-2001 Web Track. In: Information Technology: The Tenth Text Retrieval Conference, TREC 2001. NIST SP 500-250. pp.61-67.
8. Heaps, H. S.: Information Retrieval: Computational and Theoretical Aspects. Academic Press, New York, 1978.
9. Jeong, B., Omiecinski, E.: Inverted File Partitioning Schemes in Multiple Disk Systems. IEEE Transactions on Parallel and Distributed Systems, 6(2):142-153. 1995.
10. Jones, C.B., Purves, R., Ruas, A., Sanderson, M., Sester, M., van Kreveld, M., Weibel, R.: Spatial information retrieval and geographical ontologies an overview of the SPIRIT project. In Proceedings of the 25th ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 387-388. ACM Press, 2002.
11. Little, M. C.: JavaSim User's Guide. Public Release 0.3, Version 1.0. http://javasim.ncl.ac.uk/manual/javasim.pdf, University of Newcastle upon Tyne, 2001.
12. Lu, Z., McKinley, K.: Partial collection replication versus caching for information retrieval systems. In Proceedings of the ACM International Conference on Research and Development in Information Retrieval, pp: 248-255. 2000.
13. Ribeiro-Neto, B. Barbosa, R.: Query performance for tightly coupled distributed digital libraries. In Proceedings of the 3rd ACM Conference on Digital Libraries, pp: 182-190. 1998.
14. Spink, A., Jansen, B. J., Wolfram, D., Saracevic, T.: From e-sex to e-commerce: Web search changes. IEEE Computer 35(3): 107-109. 2002.
15. Tomasic, Al, Garcia-Molina, H.: Performance of inverted indices in shared-nothing distributed text document information retrieval systems. In Proceedings of the 2nd International Conference on Parallel and Distributed Information Systems, pp: 8-17, 1993.

Applying the Divergence from Randomness Approach for Content-Only Search in XML Documents

Mohammad Abolhassani and Norbert Fuhr

Institute of Informatics and Interactive Systems, University of Duisburg-Essen,
47048 Duisburg, Germany
{mohasani,fuhr}@is.informatik.uni-duisburg.de

Abstract. Content-only retrieval of XML documents deals with the problem of locating the smallest XML elements that satisfy the query. In this paper, we investigate the application of a specific language model for this task, namely Amati's approach of divergence from randomness. First, we investigate different ways for applying this model without modification by redefining the concept of an (atomic) document for the XML setting. However, this approach yields a retrieval quality lower than the best method known before. We improved the retrieval quality through extending the basic model by an additional factor that refers to the hierarchical structure of XML documents.[1]

1 Introduction

As XML document collections become more and more available, there is a growing need for retrieval methods exploiting the specific features of this type of documents. Since XML documents contain explicit information about their logical structure, XML retrieval methods should take into account the structural properties of the documents to be retrieved. One of the two tracks of INEX (initiative for the evaluation of XML retrieval [6]) deals with *content-only queries,* where only the requested content is specified. Instead of retrieving whole documents, the IR system should aim at selecting document components that fulfil the information need. Following the FERMI model [3], these components should be the deepest components in the document structure, i. e. most specific, while remaining exhaustive to the information need.

Whereas classical IR models have treated documents as atomic units, XML markup implies a tree-like structure of documents. Content-only queries now search for subtrees of minimum size that are relevant to the query. In order to address this problem, most approaches are based on the notion of the so-called *index nodes* (or index elements): Given the XML markup, not every XML element should be considered as a possible answer, e.g. because the element is too fine-grained or it is missing important elements, like a section body without the section title. So first the set of index nodes has to be defined in some way, e.g. based on the DTD, or document-specific by applying some heuristics. Now there are two possible approaches for addressing the retrieval task:

[1] The work presented in this paper is founded by the German Research Foundation (DFG), as part of the CLASSIX project.

S. McDonald and J. Tait (Eds.): ECIR 2004, LNCS 2997, pp. 409–419, 2004.

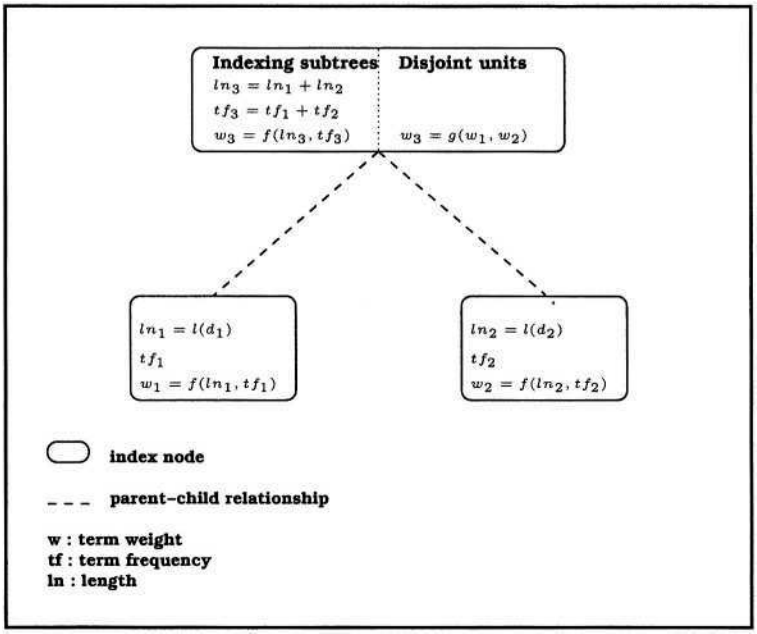


Fig. 1. Approaches for computing the indexing weights of inner nodes

Indexing subtrees: The complete text of any index node is treated like an atomic document, and some standard indexing method is applied. Due to the hierarchical structure, index nodes may be contained within each other. In contrast, most indexing methods assume that the collection consists of disjoint text blocks, and so care has to be taken in order to avoid violation of this assumption. Furthermore, the overlapping leads to some redundancy in the index file. Piwowarski et al. propose a Bayesian network approach, where the retrieval weight of an index node also depends on the weights of those nodes in which it is contained [11]. Grabs and Schek apply this idea when the query also involves structural conditions, regarding as collection only those XML elements which are fulfilling the structural conditions [9].

Disjoint units: The document is split into disjoint units, such that the text of each index node is the union of one or more of these disjoint parts. Then standard indexing methods can be applied to the disjoint units, by treating them like atomic documents, where the collection is made up of the units of the documents in the collection. For retrieval, indexing weights for nodes consisting of several units must be aggregated in some way; this makes the retrieval process more complex. Ogilvie and Callan describe a language model following this approach, where the language models of a 'higher level' node is computed as the weighted sum of the language models of its units [10].

Figure 1 illustrates the differences between the two approaches for an example document: The subtree method first collects word statistics (like e.g. document length *ln*, within-document frequency *tf*) for the complete text contained in the subtree, and then computes the indexing weight w based on these statistics. In contrast, the disjoint units method first computes indexing weights for the leaf nodes, whereas the weights for the inner nodes are derived from the combination of the weights in the leaf nodes.

Fuhr and Grossjohann describe an augmentation model based on the disjoint units approach [4]. Here indexing weights of units are propagated to the index nodes containing these units. However, when propagating from one index node to the next comprising node, the indexing weights are downweighted by multiplying them with a so-called augmentation factor. The experimental evaluation within INEX [8] showed that this approach leads to top performance among the participating systems. However, the augmentation model makes no assumptions about the underlying indexing model. For the INEX runs, we used the BM25 indexing formula.

In this paper, we present a new model for content-only retrieval which combines the subtree approach with language models. As starting point, we chose Amati's framework of retrieval, called *Divergence From Randomness (DFR)* [1,2]. We investigate several possibilities for applying this approach to XML retrieval, and combine it also with ideas from the augmentation approach.

The remainder of this paper is structured as follows: First in Section 2 we give a brief survey into Amati's model. Then we investigate the application of this approach to XML retrieval in Section 3. Finally, in Section 4, we give a summary and an outlook on our future work.

2 Divergence from Randomness

Amati and Rijsbergen introduce a framework for deriving probabilistic models of IR [1]. These models are non-parametric models of IR as obtained in the *language model* approach. The term weighting models are derived by measuring the divergence of the actual term distribution from that obtained under a random process.

There are two basic assumptions underlying this approach:

1. Words which bring little information are randomly distributed on the whole set of documents. One can provide different basic probabilistic models, with probability distribution $Prob_1$, that define the notion of *randomness in the context of IR.*
2. If one restrict statistics to the set of all documents in which a term occurs, the "elite" set, then one can derive a new probability $Prob_2$ of the occurrence of the word within a document with respect to its elite set.

Based on these ideas, the weighting formula for a term in a document is the product of the following two factors:

1. $Prob_1$ is used for measuring the *information content* of the term in a document, and $(-\log_2 Prob_1)$ gives the corresponding amount of information.
2. $Prob_2$ is used for measuring the *information gain* of the term with respect to its 'elite' set (the set of all documents in which the term occurs). The less the term is expected in a document with respect to its frequency in the elite set, measured by the counter-probability $(1 - Prob_2)$, the more the amount of information is gained with this term.

Now the weight of a term in a document is defined as

$$w = (1 - Prob_2) \cdot (-\log_2 Prob_1) = Inf_2 \cdot Inf_1 \tag{1}$$

For computing the two probabilities, the following parameters are used:

N number of documents in the collection,
tf term frequency within the document (since different normalisations are applied to the term frequency, we use tf_1 and tf_2 in the following formulas),
n size of the elite set of the term,
F term frequency in elite set.

Furthermore, let $\lambda = F/N$ in the following.

As probability distribution for estimating $Prob_1$, three different probabilistic models are regarded in [1]; using various approximations, this finally leads to seven different formulas. In this paper, we use only two of them:

D The approximation of the binomial model with the divergence:

$$Inf_1 = tf_1 \cdot \log_2 \frac{tf_1}{\lambda} + \left(\lambda + \frac{1}{12tf_1} - tf_1\right) \cdot \log_2 e + 0.5 \log_2(2\pi \cdot tf_1) \tag{2}$$

G The Geometric as limiting form of the Bose-Einstein model:

$$Inf_1 = -log_2 \frac{1}{1+\lambda} - tf_1 \cdot \log_2 \frac{\lambda}{1+\lambda} \tag{3}$$

For the parameter $Inf_2 = (1 - Prob_2)$ (which is also called *first normalisation*), $Prob_2$ is defined as the probability of observing another occurrence of the term in the document, given that we have seen already *tf* occurrences. For this purpose, Amati regards two approaches:

L Based on Laplace's law of succession, he gets

$$Inf_2 = \frac{1}{tf_2 + 1} \tag{4}$$

B Regarding the ratio of two Bernoulli processes yields

$$Inf_2 = \frac{F + 1}{n \cdot (tf_2 + 1)} \tag{5}$$

These parameters do not yet consider the length of the document to be indexed. For the relationship between document length and term frequency, Amati regards two alternative hypotheses concerning the the densityfunction $\rho(l)$ of the term frequency in the document (where c is a constant to be chosen):

H1 The distribution of a term is uniform in the document. The term frequency density $\rho(l)$ is constant; that is $\rho(l) = c$.
H2 The term frequency density $\rho(l)$ is a decreasing function of the length; that is $\rho(l) = c/l$.

In this paper, we also regard the generalisation of these two assumptions:

$$\rho(l) = c \cdot l^{\beta} \tag{6}$$

where β is a parameter to be chosen (we get H1 with $\beta = 0$ and H2 with $\beta = -1$)

In order to consider length normalisation, Amati maps the *tf* frequency onto a normalised frequency *tfn* computed in the following way: Let $l(d)$ denote the length of document d and *avl* is the average length of a document in the collection. Then *tfn* is defined as:

$$tfn = \int_{l(d)}^{l(d)+avl} \rho(l) dl \tag{7}$$

This approach yields $tfn = tf \cdot \frac{avl}{l(d)}$ for H1 and $tfn = tf \cdot \log_2(1 + \frac{avl}{l(d)})$ for H2.

For considering these normalisations, Amati sets $tf_1 = tf_2 = tfn$ in formulas 2–5 and then computes the term weight according to eqn 1.

For retrieval, the query term weight *qtf* is set to the number of occurrences of the term in the query. Then a linear retrieval function is applied:

$$R(q,d) = \sum_{t \in q} qtf \cdot Inf_2(tf_2) \cdot Inf_1(tf_1) \tag{8}$$

3 Applying Divergence from Randomness to XML Documents

3.1 Test Setting

For our experiments, we used the INEX test collection [6]. The document collection is made up of the full-texts, marked up in XML, of 12107 articles of the IEEE Computer Society's publications from 12 magazines and 6 transactions, covering the period of 1995–2002, and totalling 494 megabytes in size. Although the collection is relatively small compared with TREC, it has a suitably complex XML structure (192 different content models in DTD) and contains scientific articles of varying length. On average an article contains 1532 XML nodes, where the average depth of a node is 6.9 (a more detailed summary can be found in [5]). For our experiments, we defined four levels of index nodes for this DTD, where the following XML elements formed the roots of index nodes: article, section, ss1, ss2 (the latter two elements denote two levels of subsections).

As queries, we used the 24 content-only queries from INEX 2002 for which relevance judgements are available. Figure 2 shows the DTD of the queries applied. As query terms, we considered all single words from the topic title, the description and the keywords section.

For relevance assessments, INEX uses a two-dimensional, multi-valued scale for judging about relevance and coverage of an answer element. In our evaluations described below, recall and precision figures are based on a mapping of this scale onto a one-dimensional, binary scale where only the combination 'fully relevant'/'exact coverage' is treated as relevant and all other combinations as non-relevant. For each query, we considered the top ranking 1000 answer elements in evaluation.[2].

[2] The official INEX 2002 evaluation was based on the top 100 elements only — for details of the INEX evaluation, see [7].

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- An inex_topic has 4 parts: title, description, narrative and
keywords, and 3 attributes: the official INEX topic-id, query type
(CO=content-only, CAS=content-and-structure),
and ct-no (candidate topic number) -->

<!ELEMENT inex_topic  (title,description,narrative,keywords)>
<!ATTLIST inex_topic
topic_id   CDATA  #REQUIRED
  query_type CDATA  #REQUIRED
  ct_no      CDATA  #REQUIRED
>
<!ELEMENT title (#PCDATA)>
<!ELEMENT description   (#PCDATA)>
<!ELEMENT narrative     (#PCDATA)>
<!ELEMENT keywords      (#PCDATA)>
```

Fig.2. DTD of the INEX 2002 queries

3.2 Direct Application of Amati's Model

In Section 2, we have described the basic model along with a subset of the weighting functions proposed by Amati. Given that we have two different formulas for computing Inf_1 as well as two different ways for computing Inf_2, we have four basic weighting formulas which we are considering in the following.

In a first round of experiments, we tried to apply Amati's model without major changes. However, whereas Amati's model was defined for a set of atomic documents, CO retrieval is searching for so-called *index nodes,* i.e. XML elements that are meaningful units for being returned as retrieval answer.

As starting point, we assumed that the complete collection consists of the concatenation of all XML documents. When we regard a single index node, we assume that the complete collection consists of documents having the same size as our current node. Let L denote the total length of the collection and $l(d)$ the length of the current node (as above), then we compute the number of hypothetical documents as $N = L/l(d)$.

Table 1 shows the experimental results. The first two result columns show the average precision values for this setting when applying the four different weighting functions. We assume that the poor performance is due to the fact that the weights derived from different document lengths are not comparable.

As an alternative method, we computed the average size of an index node. The two last columns in Table 1 show a much better retrieval quality for this case.

In the subsequent experiments, we focused on the second approach. By referring to the average size of an index node we were also able to apply document length normalisation according to Equation 5. In conformance with H1 and H2 (explained in Section 2) we tried the values 0 and -1 for β. The two first and two last (result) columns of Table 2 show the corresponding results. The results show that length normalisation with $\beta = -1$ improves retrieval quality in most cases. These results were also in conformance with Amati's findings that $\beta = -1$ gives better results than $\beta = 0$.

Table 1. Results from direct application vs. augmentation approach

document length	Dynamic		Fixed	
	B Norm.	L Norm.	B Norm.	L Norm.
Bernoulli	0.0109	0.0356	0.0640	0.0717
Bose-Einstein	0.0214	0.0338	0.0468	0.0606
Augmentation	0.1120			

Table 2. Results from 2nd normalisation with four different values for β

	$\beta = 0$		$\beta = -0.75$		$\beta = -0.80$		$\beta = -1$	
	B Norm.	L Norm.	B Norm.	L Norm.	B Norm.	L Norm.	B Norm.	L Norm.
Bernoulli	0.0391	0.0586	0.0799	0.1026	0.0768	0.1005	0.0640	0.0900
Bose-Einstein	0.0376	0.0609	0.0453	0.0653	0.0448	0.0654	0.0376	0.0651

Subsequently we tried some other values for β. The four middle (result) columns of Table 2 show the corresponding results for $\beta = -0.75$ and $\beta = -0.80$, with which we got better results.

Overall, using a fixed average document length, and length normalisation, gave better results than those achieved in the first round. However, the resulting retrieval quality was still lower than that of the augmentation approach (see Table 1). Thus, in order to arrive at a better retrieval quality, we investigated other ways than straightforward application of Amati's model.

3.3 Considering the Hierarchical Structure of XML Documents

In order to consider the hierarchical structure of our documents, we investigated different ways for incorporating structural parameters within the weighting formula. Considering the basic ideas, as described in Section 2, the most appropriate way seemed the modification of the Inf_2 parameter, which refers to the 'elite' set. Therefore, we computed Inf_1 as above, by performing document length normalisation with respect to the average size of an index node.

For computing Inf_2, we also applied document length normalisation first, thus yielding a normalised term frequency *tfn*. Then we investigated several methods for 'normalising' this factor with respect to the hierarchical document structure; we call this process *third normalisation*. For this purpose, we introduced an additional parameter $h(d)$ specifying the height (or level) of an index node relative to the root node (which has $h = 1$).

Using the level information, we first tried several heuristic formulas like $tf_2 = tfn \cdot h(d)^{\alpha}$ and $tf_2 = tfn \cdot h(d)^{-\alpha}$, which, however, did not result in any improvements. Finally, we came up with the following formula:

$$tf_2 = tfn \cdot (h(d)/\alpha) \tag{9}$$

Here α is a constant to be chosen, for which we tried several values. However, the experiments showed that the choice of α is not critical.

Table 3. Average precision for the Bose-Einstein L Norm combination with various values of α

α	2	4	9	16	20	32	64	96	104	116	128
prec.	0.0726	0.0865	0.0989	0.1059	0.1077	0.1083	0.1089	0.1094	0.1087	0.1081	0.1077

Table 3 shows the results for the combination of Bose-Einstein and Laplace normalisation, for which we got significant improvements. This variant also gave better results in Amati's experiments.

The significant benefits through the third normalisation are also confirmed by the recall-precision curves shown in Figure 3, where we compare our results (DFR) with and without third normalisation to that of the augmentation approach.

In INEX 2003[3] we used the best configuration according to our experimental results, i.e. Bose-Einstein and L Normalisation with the parameters $\alpha = 96$ and $\beta = -0.80$. This submission ranked high among all submissions. The average precision achieved was 0.0906, while we got 0.1010 through our "augmentation" method with 0.2 as "augmentation factor". Figure 4 shows the recall-precision curves for these two submissions, confirming again that DFR with third normalisation performs almost as well as the augmentation approach.

In order to explain the effect of third normalisation, let us consider the weighting formula for Inf_2 again; we are using the Laplace version of this formula, which yields:

$$Inf_2 = \frac{1}{tfn + 1} \tag{10}$$

Using third normalisation, we now have instead:

$$Inf_2 = \frac{1}{\frac{h(d)}{\alpha} \cdot tfn + 1} \tag{11}$$

In the latter formula, α controls the influence of the level parameter; for $\alpha = 1$ and $h(d) = 1$, we would get the same results as before.

As described by Amati, Inf_2 measures the 'risk' or (potential) gain if the term is accepted as a descriptor of the document. In both formulas, the gain increases as *tf* decreases. However, there are two major differences:

1. In general, third normalisation yields a higher gain, since we got the best retrieval performance for values of the constant α which are much higher than those of the level $h(d)$.
2. The risk/gain is higher for smaller levels. This observation conforms to the general goal of the CO queries of the INEX task, where the most specific answers (i.e. those with higher levels) should be preferred. Thus, if the system returns a lower level element, the risk is higher.

4 Conclusions and Outlook

XML retrieval is an important new area for the application of IR methods. Whereas little research has been performed on retrieval of structured documents in the past,

[3] http://inex.is.informatik.uni-duisburg.de:2003/

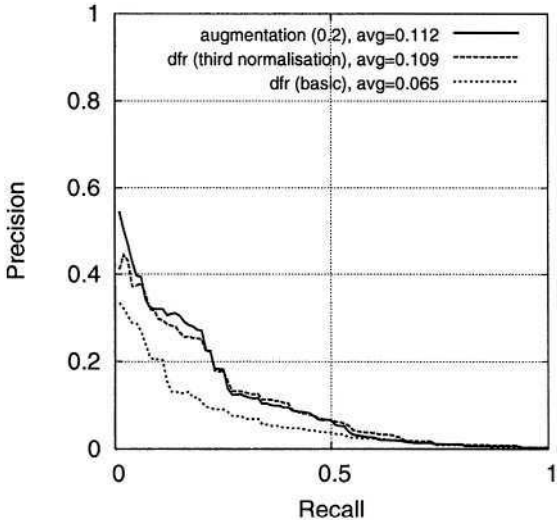


Fig. 3. Recall-Precision curves for the best approaches (INEX 2002)

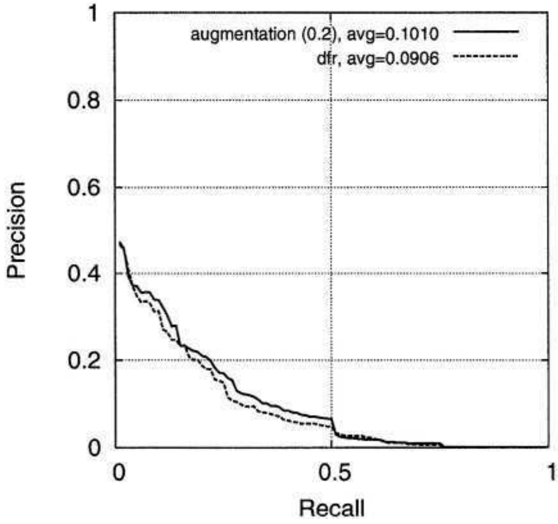


Fig. 4. Recall-Precision curves for the INEX 2003 queries

the increasing availability of XML collections offers the opportunity for developing appropriate retrieval methods. Content-only retrieval of XML documents corresponds to the classic ad-hoc retrieval task of atomic documents, but with the additional constraint of locating the smallest XML elements that satisfy the query.

In this paper, we have investigated the application of a language model approach for content-only retrieval. We have shown that a straightforward application of language models is possible by appropriate redefinition of the concept of an (atomic) document for the XML setting. In this setting, the experimental results for the different weighting formulas are in line with Amati's findings for the TREC collection. However, the retrieval quality resulting from this direct application was lower than the best results from the first round of INEX.

By adopting ideas from the successful augmentation approach, we have extended Amati's model by a third normalisation component which takes into account the hierarchical structure of XML documents. This approach has improved results, thus leading to a retrieval quality comparable to that of the augmentation approach.

We view our work as a starting point for developing appropriate language models for XML retrieval. In this paper, we have considered only one specific model of this kind, for which we were able to provide an extension yielding a high retrieval quality. In the future, we will study also other language models and investigate different extensions for coping with XML retrieval.

References

[1] Amati, G.; van Rijsbergen, C. (2002). Probabilistic models of Information Retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems 20(4),* pages 357–389.

[2] Amati, G. (2003). *Probability Models for Information Retrieval based on Divergence from Randomness.* PhD thesis, University of Glasgow.

[3] Chiaramella, Y.; Mulhem, P.; Fourel, F. (1996). *A Model for Multimedia Information Retrieval.* Technical report, FERMI ESPRIT BRA 8134, University of Glasgow.

[4] Fuhr, N.; Großjohann, K. (2001). XIRQL: A Query Language for Information Retrieval in XML Documents. In: Croft, W.; Harper, D.; Kraft, D.; Zobel, J. (eds.): *Proceedings of the 24th Annual International Conference on Research and development in Information Retrieval,* pages 172–180. ACM, New York.

[5] Fuhr, N.; Gövert, N.; Kazai, G.; Lalmas, M. (2002). INEX: INitiative for the Evaluation of XML Retrieval. In: Baeza-Yates, R.; Fuhr, N.; Maarek, Y. S. (eds.): *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval.*
`http://www.is.informatik.uni-duisburg.de/bib/xml/Fuhr_etal_02a.html.`

[6] Fuhr, N.; Gövert, N.; Kazai, G.; Lalmas, M. (eds.) (2003). *INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the First INEX Workshop. Dagstuhl, Germany, December 8–11, 2002,* ERCIM Workshop Proceedings, Sophia Antipolis, France. ERCIM.
`http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf.`

[7] Gövert, N.; Kazai, G. (2003). Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2002. In [6], pages 1–17.
`http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf.`

[8] Gövert, N.; Fuhr, N.; Abolhassani, M.; Großjohann, K. (2003). Content-oriented XML retrieval with HyREX. In [6], pages 26–32.
`http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf.`

[9] Grabs, T.; Schek, H.-J. (2003). Flexible Information Retrieval from XML with PowerDB-XML. In [6], pages 141–148.
`http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf.`

[10] Ogilvie, P.; Callan, J. (2003). Language Models and Structure Document Retrieval. In [6], pages 33–40.
http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf.

[11] Piwowarski, B.; Faure, G.-E.; Gallinari, P. (2003). Bayesian Networks and INEX. In [6], pages 149–154.
http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf.

This page intentionally left blank

Author Index