Markus Ackermann Bettina Berendt
Marko Grobelnik Andreas Hotho
Dunja Mladenič Giovanni Semeraro
Myra Spiliopoulou Gerd Stumme
Vojtěch Svátek Maarten van Someren (Eds.)

LNAI 4289

Semantics, Web and Mining

Joint International Workshops, EWMF 2005 and KDO 2005
Porto, Portugal, October 2005
Revised Selected Papers

Springer

Lecture Notes in Artificial Intelligence 4289

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Markus Ackermann Bettina Berendt
Marko Grobelnik Andreas Hotho
Dunja Mladenič Giovanni Semeraro
Myra Spiliopoulou Gerd Stumme
Vojtěch Svátek Maarten van Someren (Eds.)

Semantics, Web and Mining

Joint International Workshops, EWMF 2005 and KDO 2005
Porto, Portugal, October 3 and 7, 2005
Revised Selected Papers

Springer

Volume Editors

Markus Ackermann
University of Leipzig, E-mail: markus.ackermann@rz.uni-leipzig.de

Bettina Berendt
Humboldt University Berlin, E-mail: berendt@wiwi.hu-berlin.de

Marko Grobelnik
J. Stefan Institute, Ljubljana, E-mail: marko.grobelnik@ijs.si

Andreas Hotho
University of Kassel, E-mail: hotho@cs.uni-kassel.de

Dunja Mladenič
J. Stefan Institute, Ljubljana, E-mail: dunja.mladenic@ijs.si

Giovanni Semeraro
University of Bari, E-mail: semeraro@di.uniba.it

Myra Spiliopoulou
Otto-von-Guericke-University Magdeburg, E-mail: myra@iti.cs.uni-magdeburg.de

Gerd Stumme
University of Kassel, E-mail: stumme@cs.uni-kassel.de

Vojtěch Svátek
University of Economics, Prague, E-mail: svatek@vse.cz

Maarten van Someren
University of Amsterdam, E-mail: maarten@science.uva.nl

Library of Congress Control Number: 2006936937

CR Subject Classification (1998): I.2, H.2.8, H.3-4, H.5.2-4, K.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-47697-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-47697-9 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11908678 06/3142 5 4 3 2 1 0

Preface

Finding knowledge – or meaning – in data is the goal of every knowledge discovery effort. Subsequent goals and questions regarding this knowledge differ among knowledge discovery (KD) projects and approaches. One central question is whether and to what extent the meaning extracted from the data is expressed in a formal way that allows not only humans but also machines to understand and re-use it, i.e., whether the semantics are formal semantics. Conversely, the input to KD processes differs between KD projects and approaches. One central question is whether the background knowledge, business understanding, etc. that the analyst employs to improve the results of KD is a set of natural-language statements, a theory in a formal language, or somewhere in between. Also, the data that are being mined can be more or less structured and/or accompanied by formal semantics.

These questions must be asked in every KD effort. Nowhere may they be more pertinent, however, than in KD from Web data ("Web mining"). This is due especially to the vast amounts and heterogeneity of data and background knowledge available for Web mining (content, link structure, and usage), and to the re-use of background knowledge and KD results over the Web as a global knowledge repository and activity space. In addition, the (Semantic) Web can serve as a publishing space for the results of knowledge discovery from other resources, especially if the whole process is underpinned by common ontologies.

We have explored this close connection in a series of workshops at the European Conference on Machine Learning / Principles and Practice of Knowledge Discovery from Databases (ECML/PKDD) conference series (Semantic Web Mining, 2001, 2002) and in the selection of papers for the post-proceedings of the European Web Mining Forum 2003 Workshop (published as the Springer LNCS volume Web Mining: From Web to Semantic Web in 2004). We have also investigated the uses of ontologies (as the most commonly used type of formal semantics) in KD in the Knowledge Discovery and Ontologies workshop in 2004.

In 2005, we organized, in two partly overlapping teams and again at ECML/-PKDD, a workshop on Web mining (European Web Mining Forum) and a workshop on Knowledge Discovery and Ontologies. The submissions, and in particular the highest-quality accepted contributions, convinced us that the specific importance of semantics for Web mining continues to hold. We therefore decided to prepare a joint publication of the best papers from the two workshops that presented a variety of ways in which semantics can be understood and brought to bear on Web data. In addition, we included a particularly fitting contribution from KDO 2004, by Vanzin and Becker. The result of our selection, the reviewers' comments, and the authors' revision and extension of their workshop papers is this book.

Paper summaries

To emphasize the common themes, we will give a combined summary of the contributions in this volume. To make it easier to understand the papers in the organizational context for which they were written and in which they were discussed, we have ordered them by workshop in the table of contents.

Understanding the Web and supporting its users was addressed in the papers of both workshops: KDO 2005 and EWMF 2005. The invited contribution of Eirinaki, Mavroeidis, Tsatsaronis, and Vazirgiannis elaborates on the role of semantics for Web personalization. Degemmis, Lops, and Semeraro concentrate on learning user profiles with help of a rich taxonomy of terms, WordNet. The subject of building ontologies and taxonomies is pursued in the papers of Bast, Dupret, Majumdar, and Piwowarski and of Fortuna, Mladenič, and Grobelnik. The former proposes a mechanism that extracts a term taxonomy from Web documents using Principal Component Analysis. Fortuna et al. present OntoGen, a tool implementing an approach to semi-automatic topic ontology construction that uses Latent Semantic Indexing and K-means clustering to discover topics from document collections, while a support vector machine is used to support the user in naming the constructed ontology concepts.

The subject of evaluating the performance of such semi-automatic ontology enhancement tools for topic discovery is studied by Spiliopoulou, Schaal, Müller, and Brunzel. Topic discovery in the Web with semantic networks is also the subject of the contribution by Kiefer, Stein, and Schlieder, who concentrate on the visibility of topics. The incorporation of semantics into the mining process is studied in the work of Svátek, Rauch, and Ralbovský on ontology-enhanced association mining, while Vanzin and Becker elaborate on the role of ontologies in interpreting Web usage patterns.

The retrieval of information from the Web is another topic that was studied in both workshops. Baeza-Yates and Poblete examine the mining of user queries made in a Web site, while Stein and Hess consider information retrieval in trust-enhanced document networks. Information retrieval from the Web is the subject of the webTopic approach proposed by Escudeiro and Jorge, who concentrate on persistent information needs that require the regular retrieval of documents on specific topics. Document classification is a further powerful means towards the same objective. The classification of Web documents is addressed by Utard and Fürnkranz, who focus on the information in hyperlinks and in the texts around them.

August 2006

EWMF and KDO 2005
Workshop Chairs

Organization

EWMF 2005 and KDO 2005 were organized as part of the 16th European Conference on Machine Learning (ECML) and the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD).

EWMF Workshop Chairs

Bettina Berendt	Institute of Information Systems Humboldt University Berlin, Germany
Andreas Hotho	Knowledge and Data Engineering Group University of Kassel, Germany
Dunja Mladenič	J. Stefan Institute Ljubljana, Slovenia
Giovanni Semeraro	Department of Informatics University of Bari, Italy
Myra Spiliopoulou	Faculty of Computer Science Otto-von-Guericke-Univ. Magdeburg, Germany
Gerd Stumme	Knowledge and Data Engineering Group University of Kassel, Germany
Maarten van Someren	Informatics Institute University of Amsterdam, Netherlands

EWMF Program Committee

Sarabjot Singh Anand	University of Warwick, UK
Mathias Bauer	DFKI, Germany
Stephan Bloehdorn	University of Karlsruhe, Germany
Janez Brank	J. Stefan Institute, Slovenia
Marko Grobelnik	J. Stefan Institute, Slovenia
Haym Hirsh	Rutgers University, USA
Ernestina Menasalvas	Universidad Politecnica de Madrid, Spain
Bamshad Mobasher	DePaul University, USA
Ion Muslea	Language Weaver, Inc., USA
Michael J. Pazzani	University of California, Irvine, USA
Lars Schmidt-Thieme	University of Freiburg, Germany
Steffen Staab	University of Koblenz-Landau, Germany

EWMF Additional Reviewers

P. Basile (University of Bari, Italy)
M. Degemmis (University of Bari, Italy)
P. Lops (University of Bari, Italy)

EWMF Sponsoring Institutions

EU Network of Excellence PASCAL
Pattern Analysis, Statistical Modelling, and Computational Learning

KDO Workshop Chairs

Markus Ackermann	Dept. of Natural Language Processing, Institute for Computer Science University of Leipzig, Germany
Bettina Berendt	Institute of Information Systems Humboldt University Berlin, Germany
Marko Grobelnik	J. Stefan Institute Ljubljana, Slovenia
Vojtěch Svátek	University of Economics Prague, Czech Republic

KDO Program Committee

Nathalie Assenac-Gilles	IRIT, Toulouse, France
Chris Biemann	University of Leipzig, Germany
Philipp Cimiano	AIFB, University of Karlsruhe, Germany
Martine Collard	University of Nice, France
Andreas Hotho	University of Kassel, Germany
François Jacquenet	University of Saint-Etienne, France
Alípio Jorge	University of Porto, Portugal
Nada Lavrač	Jožef Stefan Institute, Ljubljana, Slovenia
Bernardo Magnini	ITC-IRST, Trento, Italy
Bamshad Mobasher	DePaul University, USA
Gerhard Paaß	Fraunhofer AIS, St. Augustin, Germany
John Punin	Oracle Corporation, USA
Massimo Ruffolo	ICAR-CNR and EXEURA, Italy
Michael Sintek	DFKI, Kaiserslautern, Germany

Table of Contents

EWMF Papers

KDO Papers on KDD for Ontology

KDO Papers on Ontology for KDD

Ontology-Based Rummaging Mechanisms for the Interpretation of Web Usage Patterns

Mariângela Vanzin and Karin Becker

Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS
Av. Ipiranga, 6681, Porto Alegre, Brazil
{mvanzin, kbecker}@inf.pucrs.br

Abstract. Web Usage Mining (WUM) is the application of data mining techniques over web server logs in order to extract navigation usage patterns. Identifying the relevant and interesting patterns, and to understand what knowledge they represent in the domain is the goal of the Pattern Analysis phase, one of the phases of the WUM process. Pattern analysis is a critical phase in WUM due to two main reasons: a) mining algorithms yield a huge number of patterns; b) there is a significant semantic gap between URLs and events performed by users. In this paper, we discuss an ontology-based approach to support the analysis of sequential navigation patterns, discussing the main features of the O3R (Ontology-based Rules Retrieval and Rummaging) prototype. O3R functionality is targeted at supporting the comprehension of patterns through interactive pattern rummaging, as well as on the identification of potentially interesting ones. All functionality is based on the availability of the domain ontology, which dynamically provides meaning to URLs. The paper provides an overall view of O3R, details the rummaging functionality, and discusses preliminary results on the use of O3R.

1 Introduction

Web Mining aims at discovering insights about Web resources and their usage [1,2]. Web Usage Mining (WUM) is the application of data mining techniques to extract navigation usage patterns from records of page requests made by visitors of a Web site. Access patterns mined from Web logs may reveal useful knowledge, which can help improving the design of Web sites, analyzing users' reaction and motivation, building adaptive Web sites, improving site content, among others.

The WUM process includes the execution of specific phases [1], namely data preprocessing (used to select, clean and prepare log raw data), pattern discovery (application of data mining algorithms) and pattern analysis (evaluation of yielded patterns to seek for unknown and useful knowledge).

Pattern analysis remains a key issue in WUM area. The comprehension of mined data is difficult due to the primarily syntactic nature of web data. *Pattern interpretation* in WUM has mostly to deal with the semantic gap between URLs and events performed by users, in order to understand what usage patterns reveal in terms of site events [3]. To reduce this gap, knowledge is typically aggregated to raw data during data enrichment activities in the pre-processing phase (e.g. [4, 5]). Recent approaches

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 180 – 195, 2006.

(e.g. [3, 6,7]) investigate the contributions to WUM of domain ontologies, possibly available in the Semantic Web. Semantic Web Mining [3] is one of the trends in this direction.

Another issue is that mining techniques such as association and sequence yield a huge number of patterns, where most of them are useless, uncompressible or uninteresting to users [8]. Pattern analysts have difficulty on identifying the ones that are new and interesting for the application domain. *Pattern retrieval* deals with the difficulties involved in managing a huge set of patterns, to allow setting focus on a subset of them for further analysis.

This paper discusses the use of a domain ontology, possibly available at the Semantic Web, to support the analysis of sequential navigation patterns. The approach is based on the availability of the domain ontology, and the mapping of site URLs to ontology concepts. Pattern interpretation is performed by interactively rummaging conceptual views of mined patterns, according to different dimensions of interest (i.e. service and/or content) and abstraction levels. The domain ontology is explored to provide meaning to URLs dynamically, during the analysis phase. This contrasts with classical approaches, in which semantic enrichment is performed statically in the preprocessing phase, limiting the possibilities of analyses over the mined patterns. Pattern retrieval is addressed by filtering and clustering mechanisms, which are also based on the ontology. A preliminary version of these ideas was presented in [9].

The approach has been implemented in a prototype, called O3R (Ontology-based Rules Retrieval and Rummaging). This papers presents an overview of O3R, focusing on the rummaging functionality. Details on the filtering functionality are provided in [10]. The paper also presents preliminary results of O3R evaluation.

The remainder of this paper is structured as follows. Section 2 summarizes related work. Section 3 provides an overview of O3R, and discusses the underlying ontology and pattern representations. Section 3 details the rummaging functionality, which aims at supporting pattern interpretation. Filtering and Clustering functionalities, which are targeted at pattern retrieval, are described in sections 5 and 6, respectively. Section 7 reports preliminary experiences on the use of O3R in the domain of web-based learning environments. Conclusions and future work are addressed in Section 8.

2 Related Work

Several works address issues related to pattern analysis, which can be divided into syntactical and semantic approaches. Syntactical approaches, such as [2, 8], rely on prior beliefs, which express domain knowledge. Mining results that either support or contradict these beliefs are considered (un)interesting. In [11], a domain taxonomy is used to express pattern templates, in order to identify and analyze patterns (i.e. association rules) with specific properties. MINT [4] is a sequential mining language that allows the identification of navigation patterns according to structural, conceptual and statistical constraints that are specified in a mining query. Conceptual properties refer to metadata that was associated to URL statically, during pre-processing phase. The effectiveness of these approaches is related to the ability of previously expressing what is expected to be (un)interesting (i.e. belief, template, query) for a specific domain. Therefore, in practice, they are more useful for pattern retrieving than for interpretation.

Semantic approaches are targeted at providing meaning for mined patterns with regard to the domain. WUM patterns are often represented as a set of URLs. This type of pattern is hard to interpret because a URL does not necessarily express intuitive knowledge about an event in the site. Thus, in the WUM context, patterns analysis deals with the semantic gap between URLs and events on the domain, i.e. contents and services available at the site. Application events are defined according to the application domain, a non-trivial task that amounts to a detailed formalization of the site's business model, i.e. description of user behavior, interests and intentions.

Integrating domain knowledge into the WUM environment is essential for making pattern interpretation easier, and even to obtain better results in mining. Typically, knowledge is aggregated to raw data statically, as a result of data enrichment activities in the pre-processing phase. Berendt and Spiliopoulou [4] employ domain taxonomies for pre-processing log data, such that this knowledge is can be exploited by conceptual constraints in MINT mining queries. The usefulness of ontologies, in opposition to taxonomies, which are restricted to *is-a* relationships, has been addressed by more recent works. This trend is encouraged by advances on the Semantic Web [3]. Dai et al. [7] use the semantics about page content or structure in clustering, in order to discover domain level web usage profiles to be used in Web personalization. Again, this semantics is aggregated to raw data during pre-processing phase, and in addition, is restricted to contents and topology. Oberle et al. [6] propose a semantic log definition, where users' requests are described in semantic terms. Using ontology concepts, the multitude of user interests expressed by a visit to one page can be captured, in a process referred to as conceptual user tracking.

3 O3R: An Ontology-Based Approach for Pattern Analysis

The goal of the pattern analysis phase is to identify interesting patterns among the ones yielded by mining algorithms. The main issues one has to deal in this phase are: a) the volume of patterns yielded by some mining algorithms (e.g. association, sequence) can easily exceed the analysis capabilities of a human user; b) the output of Web mining algorithms is not suitable for human interpretation, unless proper data enrichment takes place, and c) the search for interesting patterns in WUM is mostly exploratory, in opposition to hypothesis verification.

Ontology-based Rules Retrieval and Rummaging (O3R) is an environment targeted at supporting the retrieval and interpretation of sequential navigation patterns. The striking feature of O3R is that all functionality is based on the availability of the domain ontology, composed of concepts describing domain events in different abstraction levels, into which URLs are mapped. This feature allows the retrieval and interpretation of *conceptual patterns*, i.e. patterns formed of concepts, in opposition to *physical patterns*, composed of URLs. All O3R functionality is based on direct manipulation of visual representations of conceptual patterns and ontology, thus enabling a pro-active involvement of domain users with minimal training and limited technical skills. Since the ontology makes the domain knowledge explicit, users are expected to be merely familiar to the domain. Users can explore the ontology to learn about the domain, and interpret and retrieve patterns more easily, based domain characteristics.

Pattern interpretation is addressed in O3R by pattern rummaging, which allows users to interactively explore pattern semantics, according to distinct abstraction levels and dimensions of interest. This approach enables to overcome the limitations of static semantic enrichment.

Retrieval functionality is targeted at managing a large volume of rules, as typically produced by sequential or association mining algorithms [11,12]. The basic idea is to reduce the search space for inspecting the meaning of the rules in the domain, by finding sets of related rules. Two approaches are provided by O3R: clustering and filtering. Clustering groups a set of related rules, according to a given similarity criterion, such that the analyst can deal with a smaller set of rules at a time. Filtering allows selecting rules that have specific properties. Once potentially interesting rules have been identified through one of these two retrieval mechanisms, the analyst can explore their meaning dynamically, using rummaging operations.

Current implementation of O3R is limited to *sequential patterns* extracted according to the sequential algorithm described in [13]. Navigation patterns input to O3R are extracted from a dataset resulting from a typical pre-processing phase [1], and no particular data enrichment is assumed.

3.1 Domain Events Representation

Events in a web site are roughly categorized as *service* (e.g. search) and *content* (e.g. hotel) [3]. O3R assumes the representation of domain events in two levels: conceptual and physical. Physically, events are represented by URLs. The conceptual level is represented by the domain ontology, which is used to dynamically associate meaning to web pages and user actions over pages.

Fig.1(a) depicts the ontology structure using a UML class diagram. The ontology is composed of concepts, representing either a content of a web page, or a service available through a page. Concepts are related to each other through hierarchical or property relationships. A hierarchical relationship connects a descendant concept to an ascendant one. Two types of hierarchical relationships are considered: *generalization*, in which the generalized concept is ascendant of a specialized one; and *aggregation*, in which the ascendant represents the whole assembly and the descendent represents one of its parts. Every concept has at most one ascendant. *Property* relationships represent arbitrary associations that connect a subject to an object.

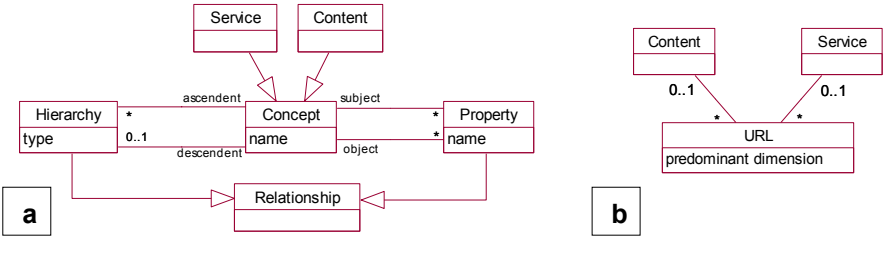


Fig. 1. Ontology structure and URL mapping

URLs are mapped into ontology concepts according to two dimensions: service and content. An URL can be mapped into one service, one content or both. When a URL is mapped into both a service and a content, it means that the URL provides a service that is closely related to some content. In that case, the mapping also defines the predominant dimension. A same ontology concept can be used in the mapping of various URLs. The above constraints are represented in Fig.1(b).

Fig.2 illustrates this ontology structure by describing the semantics of a web-based learning site. This site offers services and contents that support students learning. Services include chat, email, student's assessment, assignment submission, etc. Content is related to the material available in the site, or the subject related to some services (e.g. a forum has emails about "distance education").

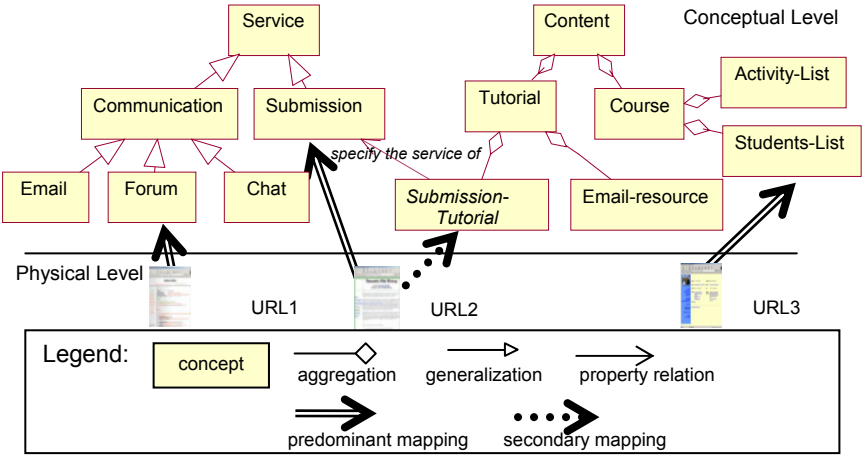


Fig. 2. Ontology and mapping examples

Fig.2 also illustrates how URLS are mapped into ontology concepts. *URL1* was mapped to the service concept *Forum*; *URL2* was mapped to both service *Submission-Tutorial* and content *Submission* concepts, where the service dimension was defined as the predominant mapping; *URL3* was mapped to the content *Student-list*.

This work does not address issues related to ontology acquisition and validation, nor mapping of the physical level into the conceptual one. We assume a domain expert is responsible for the acquisition and representation of the domain ontology, as well as for the mapping of the physical events into the corresponding conceptual ones, using manual or semi-automatic approaches, such as [6, 14]. The task of mapping URLs into ontology concepts can be laborious, but it pays off by greatly simplifying the interpretation activity, as described in the remaining of this paper. The future semantic web will contribute in reducing this effort [3].

3.2 Physical and Conceptual Patterns

The input of O3R is a set of physical patterns, i.e. sequences of URLS. Then, O3R uses the mapping between the physical and conceptual event representations to

present these patterns as a sequence of the corresponding concepts, i.e. the *conceptual patterns*. Users manipulate conceptual patterns using the provided functionality. For their analyses, users always have to establish a *dimension of interest*, which can be *content*, *service* or *content/service*. Considering the ontology of Fig.2, the physical pattern *URL1* → *URL2* corresponds to the conceptual pattern *Forum* → *Submission* according to the both service dimension and content/service dimension (where the predominant dimension is used). The pattern *URL2* → *URL3*, according to content dimension, corresponds to *Submission-Tutorial* → *Student-List*, and to *Submission* → *Student-List* according to the content/service dimension. By exploring the hierarchical relationships of the ontology, conceptual patterns at different abstraction levels can be related to a same physical pattern, as discussed in the next sections.

4 Pattern Rummaging

O3R supports interpretation activities through concept-oriented interactive pattern rummaging. It manipulates the ontology to: a) represent patterns in a more intuitive form, thus reducing the gap between URLs and site events; b) allow pattern interpretation according to different dimensions of interest and abstraction levels; c) establish different relationships between patterns. Knowledge is integrated to physical patterns dynamically, on demand, according to user's analysis interest. Fig.3(a) displays the rummaging area of O3R. Rummaging functionality is composed by the following features: a) graphical pattern representation; b) dimension of interest; c) detailing operations; d) generalized and specific patterns. These are discussed in the remaining of this section.

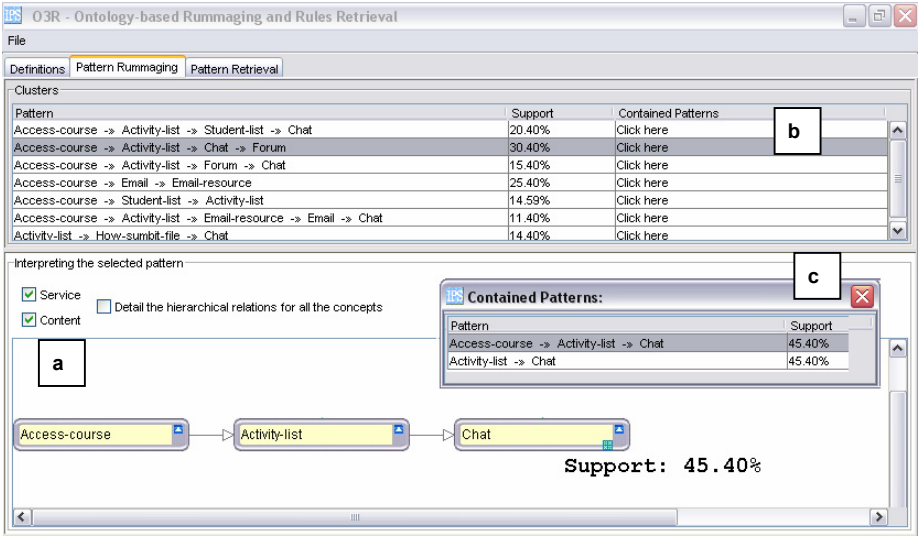


Fig. 3. Clustering and rummaging interface

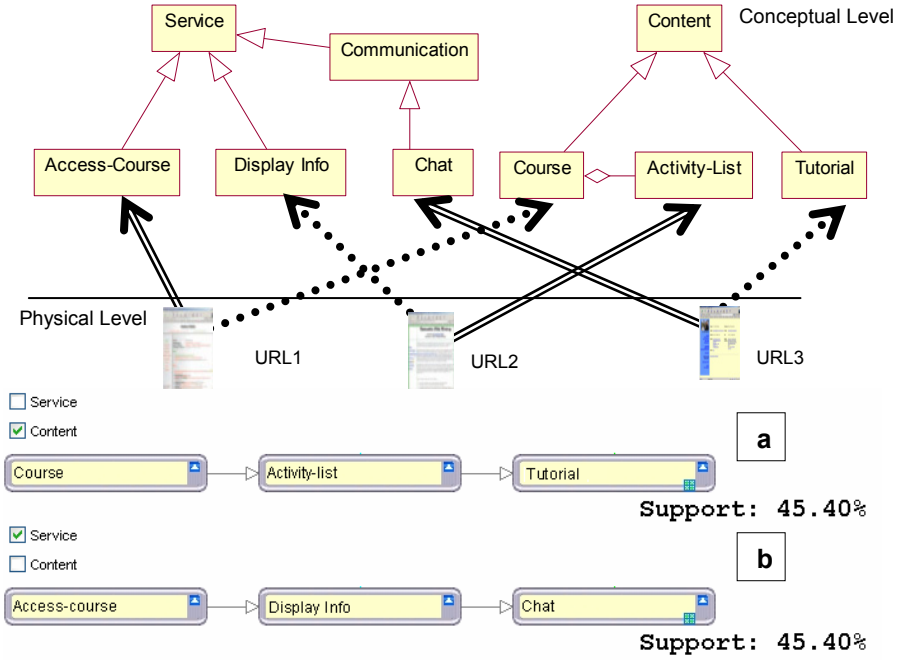


Fig. 4. Dimensions of interest

To rummage around a pattern the user has to choose a pattern and a dimension of interest. The user can select a pattern either from the Clustering area (Fig.3(b)), the Contained Patterns pop-up window (Fig.3(c)), or from the Filtering area. Filtering and clustering are discussed in sections 5 and 6, respectively. In the example of Fig.3, the user has selected the conceptual pattern *Access.course* → *Activity-list* → *Chat* from the Contained Patterns window, and the service/content dimension of interest.

By selecting a different dimension of interest, the user can dynamically interpret the same pattern differently. Consider for instance the ontology displayed in Fig.4, and the events mapping depicted. The pattern of Fig.1(a) would be displayed as in Fig.4(a) if the selected dimension of interest were content only, or as Fig.4(b), for service only.

Detailing operations allow enriching the pattern graphical representation with related concepts and respective relationships, in order to better understand pattern meaning. H*ierarchical detailing operations* dynamically include in (or remove from) the graphical representation the ascendant concept and the respective hierarchical relationship. In the example of Fig.5, using hierarchical detailing operations the user becomes aware that *Chat* is-a *Communication* tool, which is turn is-a *Service*. Likewise, he discovers that *Activity-List* is part-of *Course*, which in turn is part-of the *Content* provided by the site. Hierarchical detailing operations are triggered by clicking on the little up/down arrows displayed together with each concept.

The *property detailing operation* enables the user to interpret the pattern with the use of property relationships of the ontology, which represent arbitrary associations

that connect a subject to an object. This information is displayed in a separate window, in order to not jeopardize the graphical representation. In the example of Fig.5, the user discovers that *Chat* is about *How to submit a file*. Property detailing operation is triggered using a pop-up menu. Visually, a small cross bellow the concept indicates it is related by a property relationship.

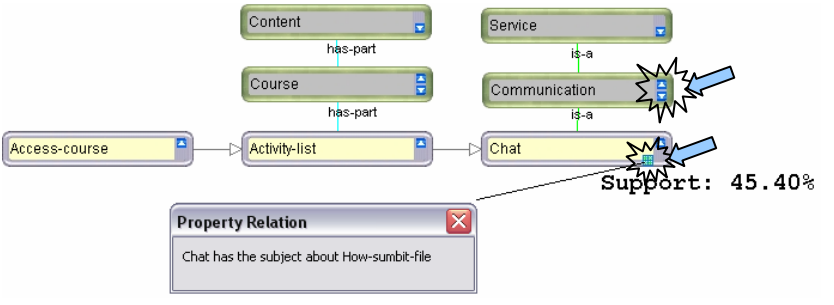


Fig. 5. Applying hierarchical and property detailing operations to a conceptual pattern

Generalized patterns are based on drill-up and drill-down operations, in an analogy to OLAP. Drill operations are a means to establish relationships among patterns in different abstraction levels dynamically. Roll-up is used to obtain a *generalized pattern*, whereas drill-down finds the specific patterns related to it. These operations explore the hierarchical relationships, i.e. specialization and aggregation.

To generalize a pattern, the user has to select the concept to be generalized, at any abstraction level, and apply the drill-up operation (double-click on the concept). For instance, in the example of Fig.5, the user could drill-up the concept *Chat* or *Communication*. Fig.6 illustrates a generalized pattern obtained by drilling up the pattern of Fig.5 (concept *Chat* was drilled up to *Communication*), with the respective support, which must be calculated from the original physical patterns. Generalized patterns are presented using different colors on the concepts to which drill-up operations were applied. Fig.6 also presents a window displaying the specific patterns found using drill-down, which is triggered by clicking in the diamond displayed together with the generalized concept. This approach for obtaining generalized patterns can be contrasted with the generation of generalized rules during the mining phase, as for example in [13], which results in the generation of a huge set of unrelated rules. In our approach, generalized rules are created on-demand, and it is always possible to relate generalized and specific rules.

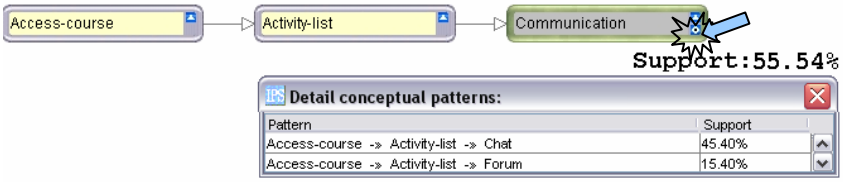


Fig. 6. Generalized and specific conceptual patterns

5 Pattern Filtering

Filtering is a useful mechanism for managing large volumes of rules. The main features of the filtering functionality are summarized in this section, and further details can be found in [10]. The filtering interface is presented in Fig.7.

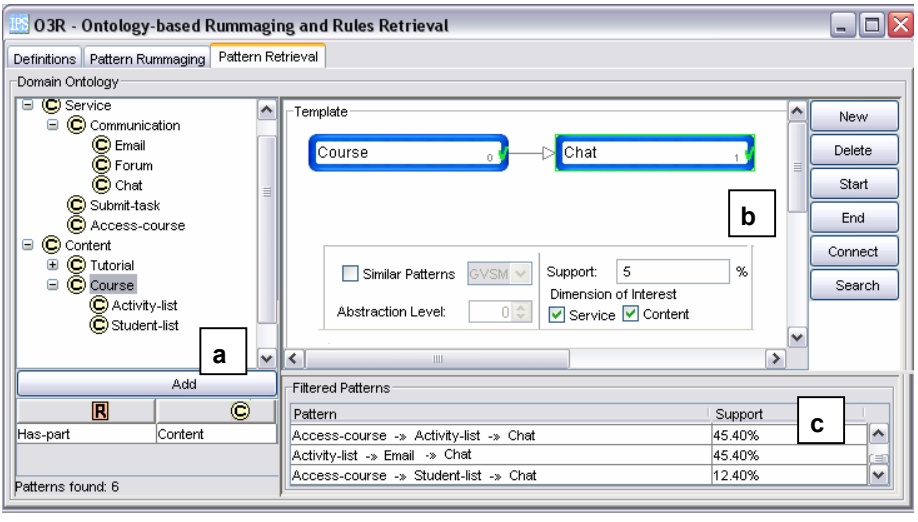


Fig. 7. Filtering Interface

In O3R, users have the support of the ontology understand the domain, and establish event-based filters that reflect potentially (un)interesting patterns. The ontology is presented in the leftmost window (Fig.7(a)), displaying concepts and their relationships. Filters are very expressive, allowing the definition of *conceptual*, *structural* and *statistical* constraints. Conceptual constraints are represented by ontology concepts, and define the interest on patterns involving specific domain events, at any abstraction level. Structural constraints establish an order among events (e.g. start with). Statistical constraints define the minimum support of sequential rules. Filters are defined visually by direct manipulation of domain concepts and structural operators (Fig.7(b)). The filter in Fig.7 defines rules involving any event of category *Course*, (immediately or not) followed by *Chat* event, with at least 5% of support.

A filtering engine examines each conceptual pattern, verifying whether it meets the statistical, conceptual and structural constraints of the filter. Two filtering engines are provided, referred to as *equivalence filtering* and *similarity filtering*. They differ on how they select rules that meet conceptual constraints. Equivalence filtering selects rules that include concepts that explicitly appear in the filter, or its descendents (i.e. more specialized concepts). Considering the example of Fig.7, all filtered patterns (Fig.7(c)) include concepts *Chat* and hierarchical descendents of *Course* (*Activity-list*, *Student-list*). On the other hand, similarity filtering considers also siblings of specified concepts, according similarity parameters specified by the user. The adopted similarity function is shown in Formula 1, where c1 e c2 are concepts, LCA is the Lowest

Common Ancestor of c1 and c2, and depth is the distance of a concept from the root of the hierarchy. The result of the similarity function is a number that ranges from 0 to 1, where Sim(c1, c2) = 1 iff c1 = c2. It is an adaptation of the similarity function proposed in [15]. Fig.8 displays an example. Unlike the filtered patterns of Fig.7(c), the filtered patterns of Fig. 8 include *Forum* and *Email* concepts, which are considered similar to *Chat* due to the common ancestor *Communication*. Patterns are displayed together with their respective similarity.

$$Sim(c_1, c_2) = \frac{2*depth\,(\mathrm{LCA}(c_1, c_2))}{depth\,(c_1) + depth\,(c_2)} \quad (1)$$

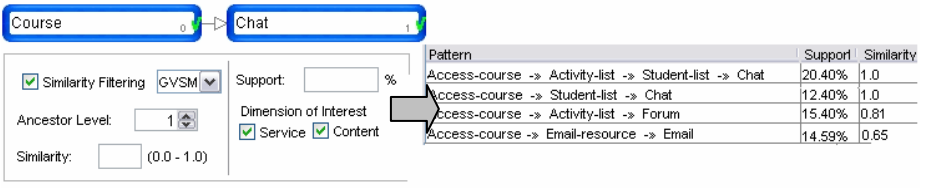


Fig. 8. Similarity filtering

6 Pattern Clustering

Clustering also is targeted at managing large amounts of rules, because it groups related rules in different sets, allowing the analyst to set focus for further inspection on a set of similar rules (or conversely, to disregard the whole set). Current implementation of O3R uses the maximal sequence rule [13] as clustering criterion. This criterion considers that each maximal sequence rule defines a cluster. Then, the maximal sequence rule and all corresponding subsequence rules are considered similar and included in the same cluster. Clustering functionality is presented in the interface together with the rummaging area (Fig.3). In the upper window (Fig.3(b)), all existing groups are displayed, where the group is represented by the maximal rule. By selecting a group, the analyst can inspect the rules it contains (Fig.3(c)) in the Contained Patterns window.

It should be stressed that other criteria are possible, and this is one of the items the user can configure in O3R (*Definitions* tab). For instance, the similarity measures proposed originally for session clustering (e.g. [16, 17]) could be adapted for this purpose.

7 Preliminary Experiences: Discovering Interesting Web Usage Patterns in a Web-Based Learning Site

O3R is currently under evaluation using a web-based site in the distance education domain. This website refers to a real course offered by PUCRS-Virtual, the distance

education department of our university. PUCRS-Virtual uses WebCT[1] to design and manage learning site. It provides tools for content propagation (e.g. texts, videos), synchronous and asynchronous communication (e.g. chat, email, forum), assignment submission, performance evaluation (e.g. quiz), among others. Our experiences refer to the analysis of navigation and learning behavior related to an intensive extracurricular course with 15 students, as represented by a web server log containing 15,953. Considering this course, two studies are used to highlight the contributions of O3R.

The first study describes the motivating scenario for O3R, which was a WUM project developed for nearly 18 months with the goal of understanding the role of WUM for discovering students' behavior. We describe the challenges faced in pattern analysis during this project, and how O3R addresses these challenges. We then establish a naïve comparison based on the opinion of the domain expert who took part in the original project. The second case study is less subjective, and was performed with the aid of students. We developed questions about the structure, content or problems of the site, and assessed whether O3R was helpful is providing correct answers.

7.1 Study 1: A Naïve Comparison with an Ad Hoc Approach

The motivating problem. In 2002-2003, we developed a project with the support of PUCRS-Virtual in order to understand the role of WUM for discovering students' behavior with regard to learning process and site usage. For this purpose, we developed a framework for analyzing page accesses in terms of the learning processes that motivate them [18]. The framework helped us to understand the mapping of the learning environment into the technological infrastructure, the specifics of the course at hand and its site, as well as WebCT functionality. Emphasis was settled on how the learning resources were distributed and accessed in the site. To deal with the semantic gap, we mapped all URLs to conceptual events. Such a mapping was developed manually. It was based on the analysis of the contents and structure of this site, additional material about the site and PUCRS Virtual pedagogic plan, as well as interviews with a domain expert. Several data mining techniques were applied during the project with the support of many tools, IBM Intelligent Miner (IM)[2] among them. This project gave us opportunity to deal in practice with most challenges inherent to pattern interpretation in WUM: an overwhelming number of rules, and the semantic gap between URLs and semantic events.

We consider in this section the case of sequential rules, which was one of the main interests of the domain expert. Considering that the goal of the project was not to produce knowledge, we limited pattern analysis to various discussions with the domain expert, which involved many meetings to exchange ideas in a period of approximately 18 months.

For this experience, we used a subset of the original log, referring to 3 days of interaction (nearly 6,000 records), which was pre-processed and enriched. This period was chosen because we knew what students were supposed to do: to study a specific subject using the materials available at the site, discuss with the classmates using the communication tools, and submit an essay using the submission functionality. Each

[1] www.webct.com/
[2] www-3.ibm.com/software/data/iminer

run of the sequential mining algorithm produced hundreds or thousands of (redundant) rules. To discuss the meaning of the rules with the expert, we decided to always pre-select a few dozens of rules that could be interesting. Based on her extensive knowledge of the course at hand and WebCT infrastructure, she would suggest possible pattern interpretations, and raise more questions, that we were suppose to answer in the next meeting. To produce these answers, most frequently we had to re-process the log to enrich it differently, and re-mine the data set.

We soon realized that we should first show more generic patterns, because they were more intuitive to the expert. When an interesting rule was identified, we would search for more specific related patterns and deep the discussion. In time, we developed a taxonomy of concepts, which was continuously validated by the expert. Hence, the taxonomy was incrementally refined. We then started to use this taxonomy to produce generalized sequential patterns with IM. In doing so, however, we had to deal with even more rules. It should be pointed out that IM does not provide adequate support for establishing relationships between a generalized rule and the corresponding specific ones.

In conclusion, the distance education department staff became excited about the results at each interaction. However, there was no available domain expert that could dedicate the time required, particularly considering the huge number of patterns.

A naïve comparison. When O3R prototype was concluded, many months later, we contacted the same domain expert to demonstrate O3R functionality and ask her opinion about the usefulness of the proposed functionality. To collect her opinion, we enacted one of our meetings to evaluate rules. We adopted the taxonomy discussed above, and enriched it with property relationships. This ontology organizes 200 concepts according to hierarchical and property relationships. We adopted exactly the same data set, and manually mapped all URLs to a concept of the ontology (content, service or both). The existing enriched datasets helped in this task. For this study, 499 URLs were mapped to domain ontology concepts. Finally, we produced sequential rules with IM, which resulted into 943 patterns.

Our demonstration session took approximately 2 hours. We operated the tool, but the domain expert instructed on what to do, after the possibilities were demonstrated. We started by showing the clusters, from which she selected one rule for rummaging. She explored the ontology relationships to detail the selected pattern, changed the dimension of interest, drilled the pattern up to generalize it, and then drilled it down to find related patterns, from which she selected another one for rummaging, and so forth. From the insight gained through rummaging, she showed interest on patterns with specific properties, which were filtered with the support of the ontology. She then selected some filtered patterns and rummaged them, leading to the definition of new filters, exploring all the interactiveness provided by O3R.

After this demonstration, we interviewed her, asking her opining about the process employed in the former experience. She pointed out that the following issues:

- the ad hoc process was very time consuming: she would spend a lot of time trying to understand what a given concept appearing in a pattern could mean, as well as what patterns could imply in practice about the learning site. Consequently, each meeting was excessively long and tiresome;

- frequently questions raised by the presented patterns implied to reprocess raw data to enriched it differently, and re-mining it. Thus, several meetings were necessary to reach a conclusion, and most questions could not be answered in the same meeting.

We than asked about her opinion on the advantages of developing the same analysis tasks with the support of O3R. She highlighted that:

- it was very easy to understand the patterns visually, using different abstraction levels and dimension of interests. She could concentrate on the tasks of understanding concepts, and how they composed patterns;
- finding interesting patterns was a consequence of pattern understanding;
- she could explore different analysis strategies, reaching unexpected patterns by the use of generalized patterns and similarity filtering;
- she could test hypothesis, which were easily represented using the ontology
- she could more easily perceive the benefits of WUM to the site evaluation and monitoring.

Finally, we presented her with a list of advantages of O3R, and asked her to sign the striking ones in her opinion, namely: interactiveness, intuitive pattern representation, visualization of patterns according to various perspectives, ability to establish various types of relationships, and support provided by domain ontology to perform analysis. She signed them all as major advantages of O3R.

The results of this study are of course very limited, in that they are very subjective and represent the opinion of a single person. Nevertheless, it is interesting to observe that O3R addressed real issues, and that its benefits were concretely perceived.

7.2 Study 2: Problem Solving Experiment

Considering the same ontology, data set and rules used for comparison in the previous study, we developed a more objective study to investigate whether the use of O3R would enable to answer questions about a learning site. We developed 5 questions that could be answered by the existing navigation rules. Table 1 summarizes the nature of each question, and relates the O3R functionality that was expected to be used to answer it. Twelve (12) subjects were asked to use O3R to answer these questions. Subjects were graduate students (master level) with some experience on KDD, and no previous contact with the learning site. As a preparation for this experiment, they attended a 30 minutes talk about WUM, and developed 5 training exercises using O3R functionality.

Fig.9 summarizes the results of this study, which cannot be further detailed here due to lack of space. As it can be seen in the graph, most users provided correct answers to the testing questions, and very few incorrect ones were provided. In most cases, the answer was considered partially correct because it involved two complementary issues, and the subjects answered only one of them. On the other hand, considering subjects' individual performance, we observed that 50% of the subjects provided correct answers to all questions, 12,5% subjects provided 5 correct answers and 12,5% correctly answered 4 questions. Considering the 50% of subjects that provided 5 or 4 correct answers, they all provided partially correct answers to the other questions. Subjects also filled in a form stating the satisfaction with regard to O3R

functionality usefulness, intuitiveness, user friendliness and overall satisfaction, using a scale [1,5]. Fig.10 displays the average score for each criterion, where 5 is the highest possible score. We are very encouraged by these preliminary results.

Table 1. Testing Questions

Id	Testing Question	Expected Functionality
T1	A problem with the structure of the site	Clustering and Rummaging
T2	A problem with the submission functionality	Custering and Rummaging
T3	Description of student behavior	Clustering and Rummaging
T4	Comparison between expected and real behavior	Filtering
T5	Comparison between expected and real behavior	Filtering
T6	Description of student behavior	Filtering and Rummaging

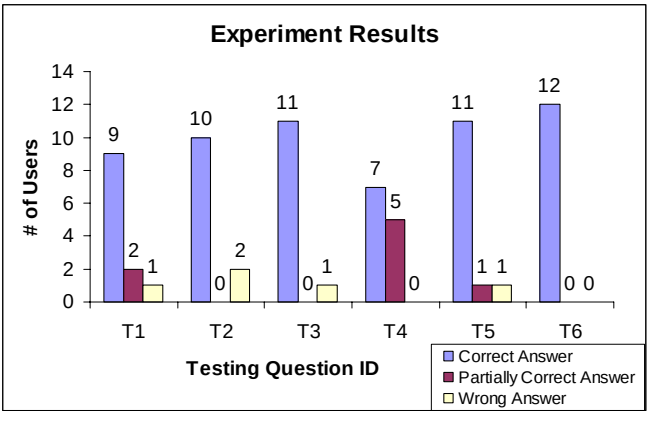


Fig. 9. Experiment results

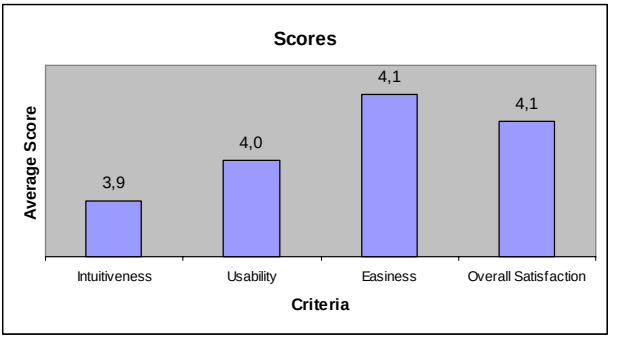


Fig. 10. O3R evaluation

8 Conclusions and Future Work

In this paper we discussed an approach that exploits domain knowledge to support pattern analysis, discussing how it is integrated in the prototype O3R. O3R is intended

to make mined patterns more easily compressible to human users, as well as to support the visual and interactive evaluation and identification of potentially interesting patterns. Functionality addresses three main problems related to patter analysis in WUM: a) a more intuitive representation patterns in order to reduce the gap between URLs and site events, b) the identification of patterns that are related to some subject of interest, the c) to identification of potentially interesting patterns through concept-oriented, interactive pattern rummaging. Grouping of patterns by different similarity criteria and visual pattern representation and manipulation complements the approach.

The prototype O3R implements the proposed approach, and preliminary experiences demonstrated a number of advantages, particularly: support for exploratory and hypothesis-based analysis; intuitive pattern representation, based on ontology concepts; easy identification of potentially interesting patterns; dynamic enrichment of data considering different dimensions of interest during the Pattern Analysis phase, without re-execution of previous phases; reduced number of rules using filtering and clustering functionalities; identification of rules with similar properties; the ability to relate generalized and specific patterns easy identification of redundant patterns through clustering usage and finally deeper insight of the domain. Of course the experiences were limited, and further work needs to be developed to soundly evaluate the contribution of O3R's features. Nevertheless, the experiences revealed a potential for problem solving and the intuitiveness of the approach. In both experiments developed, the previous training with O3R was minimal, and none of the users was experienced on WUM. In the second experiment, subjects did not even have any previous experience with the site.

O3R can be easily extended to support other mining techniques (e.g. association), as well as other algorithms for sequential patterns (e.g. [4]). Other limitations of O3R must be addressed, particularly the constraints upon ontology structure and on the semantic mapping of URLs.

Currently we are extending and evaluating O3R and studying various issues involved in the application of clustering to understand students' behavior [19]. Further research includes, among other topics, other similarity functions for clustering patterns, concept-based pattern similarity, and analyst profile learning for personalization and recommendation actions.

Acknowledgements. This work is partially supported by Fundação de Amparo à Pesquisa do Rio Grande do Sul (FAPERGS - Brazil) and the Dell/PUCRS Agreement.

References

[1] Cooley, R., Mobasher, B., and Srivastava, J. Data preparation for mining world wide web browsing patterns. Knowledge and Information Systems 1, 1 (1999), 5-32.

[2] Cooley, R. The use of web structure and content to identify subjectively interesting web usage patterns. *ACM Transactions on Internet Technology 3*, 2 (2003), 93-116.

[3] Berendt, B, Hotho, A., Stumme, G. Towards Semantic Web Mining. In: International Semantic Web Conference (2002), pp. 264-278.

[4] Berendt, B., and Spiliopoulou, M. Analysing navigation behaviour in web sites integrating multiple information systems. *The VLDB Journal, 9* (2000), 56-75. .

[5] Meo, R.; Lanzi, P.L., Matera, M. Integrating Web Conceptual Modeling and Web Usage Mining. In: WebKDD'04 (International Workshop on Web Mining and Web Usage Analysis), (2004), ACM Press.

[6] Oberle, D., Berendt, B., Hotho, A., and Gonzalez, J. Conceptual user tracking. In *International Atlantic Web Intelligence Conference* (2003), Springer, pp. 142-154.

[7] Dai, H. and Mobasher, B.. Using ontologies to discovery domain-level web usage profiles. In: 2nd *Semantic Web Mining Workshop* at ECML/PKDD-2002, (2002), ACM Press.

[8] Silberschatz, A., and Tuzhilin, A. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering 8*, 6 (1996), 970-974.

[9] Vanzin, M. and Becker, K.. Exploiting knowledge representation for pattern interpretation. In: *Workshop on Knowledge Discovery and Ontologies – KDO'04* (2004), pp. 61-71.

[10] Vanzin, M. and Becker, K. (2005). Ontology-based filtering mechanisms for web usage patterns retrieval. In: *6th International Conference on Electronic Commerce and Web Technologies - EC-Web '05* (2005), Springer-Verlag, pp. 267-277.

[11] Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I. Finding interesting rules from large sets of discovered association rules. In: *Proceedings of the third international conference on Information and knowledge management* (1994), ACM Press, pp. 401-407.

[12] Hipp, J., and Guntzer, U. Is pushing constraints deeply into the mining algorithms really what we want?: an alternative approach for association rule mining. *SIGKDD Exploration. Newsl. 4*, 1 (2002), 50-55.

[13] Agrawal, R. and Srikant, R. Mining sequential patterns. In: *11th International Conference on Data Engineering*. (1995), ACM Press, pp. 3-14.

[14] Sure, Y.; Angele, J.; Staab, S. Ontoedit: guiding ontology development by methodology and inferencing. In: *International Conference on Ontologies, Databases and Applications of Semantics (ODBASE)* (2002), pp.1205-1222.

[15] Ganesan, P., Garcia-Molina, H., and Widom, J. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems*, 21, 1 (2003), 64-93.

[16] Mobasher, B. Web Usage Mining and Personalization. In: *Practical Handbook of Internet Computing*. CRC Press, 2005.

[17] Nichele, C. and Becker, K. Clustering Web Sessions by Levels of Page Similarity. In: *10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)* (2006), Springer-Verlag, pp. 346-350.

[18] Machado, L., and Becker, K. Distance education: A web usage mining case study for the evaluation of learning sites. In: *3rd IEEE International Conference on Advantage Learning Technologies* – ICALT (2003), ACM Press, pp. 360-361.

WordNet-Based Word Sense Disambiguation for Learning User Profiles

M. Degemmis, P. Lops, and G. Semeraro

Dipartimento di Informatica - Università di Bari
Via E. Orabona, 4 - 70125 Bari - Italia
{degemmis, lops, semeraro}@di.uniba.it

Abstract. Nowadays, the amount of available information, especially on the Web and in Digital Libraries, is increasing over time. In this context, the role of user modeling and personalized information access is increasing. This paper focuses on the problem of choosing a representation of documents that can be suitable to induce concept-based user profiles as well as to support a content-based retrieval process. We propose a framework for content-based retrieval, which integrates a word sense disambiguation algorithm based on a semantic similarity measure between concepts (synsets) in the WordNet `IS-A` hierarchy, with a relevance feedback method to induce *semantic user profiles*. The document representation adopted in the framework, that we called *Bag-Of-Synsets* (BOS) extends and slightly improves the classic *Bag-Of-Words* (BOW) approach, as shown by an extensive experimental session.

1 Introduction

Due to the impressive growth of the availability of text data, there has been a growing interest in augmenting traditional information filtering and retrieval approaches with Machine Learning (ML) techniques inducing a structured model of a user's interests, the *user profile*, from text documents [13]. These methods typically require users to label documents by assigning a relevance score, and automatically infer profiles exploited in the filtering/retrieval process.

There are information access scenarios that cannot be solved through straightforward matching of queries and documents represented by keywords. For example, a user interested in retrieving "interesting news stories" cannot easily express this form of information need as a query suitable for search engines. In order to find relevant information in these problematic information scenarios, a possible solution could be to develop methods able to analyze documents the user has already deemed as interesting in order to discover relevant concepts to be stored in his personal profile. Keyword-based approaches are unable to capture the *semantics* of the user interests. They are driven by a string-matching operation: If a string is found in both the profile and the document, a match is made and the document is considered as relevant. String matching suffers from problems of *polysemy*, the presence of multiple meanings for one word, and *synonymy*, multiple words having the same meaning. Due to synonymy, relevant

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 18–33, 2006.

information might be missed if the profile does not contain the exact keywords occurring in the documents, while wrong documents might be deemed as relevant because of the occurrence of words with multiple meanings.

These problems call for alternative methods able to learn more accurate profiles that capture concepts expressing users' interests from relevant documents.

These *semantic* profiles will contain references to concepts defined in lexicons or, in a further step, ontologies. This paper proposes a framework for content-based retrieval integrating a word sense disambiguation (WSD) strategy based on WordNet with a relevance feedback method to induce *semantic user profiles* [7]. The paper is organized as follows: Section 2 presents the task of learning user profiles as a text categorization problem, Section 3 and 4 propose a strategy based on WordNet to represent documents and describe how this representation can be exploited by a relevance feedback method to learn semantic user profiles, whose effectiveness is evaluated in Section 5. Conclusions are in Section 6.

2 Learning User Profiles as a Text Categorization Problem

The content-based paradigm for information filtering is analogous to the relevance feedback in information retrieval [17], which adapts the query vector by iteratively absorbing user judgments on newly returned documents. In information filtering the tuned query vector is a profile model that specifies both keywords and their informative power. The relevance of a new item is measured by computing a similarity measure between the query vector and the feature vector representing the item. ML techniques generate a model that will predict whether a new item is likely to be of interest, based on information previously labeled by the user. ML techniques generally used are those well-suited for text categorization(TC): an inductive process automatically builds a text classifier by learning features of the categories [20]. We consider the problem of learning user profiles as a binary TC task: each document has to be classified as interesting or not with respect to user preferences. The set of categories is restricted to c_+, representing the positive class (user-likes), and c_-, the negative one (user-dislikes). We present a relevance feedback method able to learn profiles for content-based filtering. The accuracy of the keyword-based profiles inferred by this method will be compared to the accuracy of profiles learned by the same method using documents indexed by WordNet.

2.1 Document Representation

In the classical *bag of words* (BOW) model, each feature corresponds to a single word in the training set. We propose a *bag of synsets* model (BOS) in which each document is encoded as a synset vector instead of as a word vector. The task of WSD consists in deciding which of the senses of an ambiguous word is invoked in a particular use of the word [11]. As for sense repository, we adopted WordNet [7], in which nouns, verbs, adjectives and adverbs are organized into

*synset*s (*syn*onym *set*s), each representing one lexical concept. Synsets are linked by different relations (*is-a*, *part-of*, etc.) and organized in hierarchies. The main advantage of the BOS representation is that synonym words belonging to the same synset can contribute to the user profile definition by referring to the same concept. A WSD procedure reduces classification errors due to ambiguous words, allowing a better precision. We addressed the WSD problem by proposing an algorithm based on semantic similarity between synsets. In our application scenario, documents are movie descriptions represented by *slots*. Each slot is a textual field corresponding to a specific movie feature: *title*, *cast*, *director*, *summary* and *keywords*. The text in each slot is represented by the BOS model by counting separately the occurrences of a synset in the slots in which it appears.

More formally, assume that we have a collection of N documents. Let m be the index of the slot, for $n = 1, 2, ..., N$, the n-th document is reduced to five bags of synsets, one for each slot:

$$d_n^m = \langle t_{n1}^m, t_{n2}^m, \ldots, t_{nD_{nm}}^m \rangle$$

where t_{nk}^m is the k-th synset in slot s_m of document d_n and D_{nm} is the total number of synsets appearing in the m-th slot of document d_n. For all n, k and m, $t_{nk}^m \in V_m$, which is the vocabulary for the slot s_m (the set of all different synsets found in slot s_m). Document d_n is finally represented in the vector space by five synset-frequency vectors:

$$f_n^m = \langle w_{n1}^m, w_{n2}^m, \ldots, w_{nD_{nm}}^m \rangle$$

where w_{nk}^m is the weight of the synset t_k in the slot s_m of document d_n and can be computed in different ways: It can be simply the number of times synset t_k appears in slot s_m or a more complex TF-IDF score. The strategy we adopted to weight synsets is described in Section 4.1.

2.2 Related Work

Our work was mainly inspired by:

- *Syskill & Webert* [15], that suggests to learn user profiles as Bayesian classifiers;
- *ifWeb* [1], that supports users in document searching by maintaining user profiles which store both interests and explicit *dis*interests;
- *SiteIF* [10], which exploits a sense-based representation to build a user profile as a semantic network whose nodes represent senses of the words in documents requested by the user;
- *Fab* [2], which adopts a Rocchio [17] relevance feedback method to create and update the user personal model (selection agent) that are directly compared to determine similar users for collaborative recommendations.

According to these successful works, we conceived the content-based system presented in this work as a text classifier able 1) to deal with a sense-based document representation and 2) to distinguish between interests and *dis*interests of

users. The strategy we propose to shift from a keyword-based document representation to a sense-based document representation is *to integrate lexical knowledge in the indexing step of training documents*. Several methods have been proposed to accomplish this task. In [18], WordNet is used to enhance neural network learning algorithms. This approach makes use of synonymy alone and involves a manual word sense disambiguation (WSD) step, whereas this paper exploits both synonymy and hypernymy and is completely automatic. Scott and Matwin proposed to include WordNet information at the feature level by expanding each word in the training set with *all* the synonyms for it in WordNet, including those available for each sense, in order to avoid a WSD process [19]. This approach has shown a decrease of effectiveness in the obtained classifier, mostly due to the word ambiguity problem. The work by Scott and Matwin suggests that some kind of disambiguation is required. Subsequent works tried to investigate whether embedding WSD in document classification tasks improves classification accuracy. Hotho and Stumme used WordNet-based WSD and feature weighting to achieve improvements of clustering results: They showed beneficial effects when background knowledge stored in WordNet is included into text clustering [8]. Bloedhorn and Hotho compared three strategies to map words to senses: No WSD, most frequent sense as provided by WordNet, WSD based on context [3]. They found positive results on the Reuters 25178, the OSHUMED and the FAODOC corpus. In [21], a WSD algorithm based on the general concept of Extended Gloss Overlaps is used and classification is performed by a Support Vector Machine classifier applied to the two largest categories of the Reuters 25178 corpus and two Internet Movie Database movie genres[1]. The relevant outcome of this work is that, when the training set is small, the use of WordNet senses combined with words improves the performance of the classifier. Also in a more recent work [12], the authors provided a sound experimental evidence of the quality of their approach for embedding WSD in classification tasks, especially when the training sets are small.

3 A WordNet-Based Algorithm for Word Sense Disambiguation

The goal of a WSD algorithm is to associate the most appropriate meaning or sense s to a word w in document d, by exploiting its *window of context* (or more simply *context*) C, that is a set of words that precede and follow w. The sense s is selected from a predefined set of possibilities, usually known as *sense inventory*. In the proposed algorithm, the sense inventory is obtained from WordNet. For example, let us consider the document d: "The white cat is hunting the mouse". The text in d is processed by two basic phases: (a) tokenization, part-of-speech tagging (POS) and lemmatization; (b) synset identification by WSD. Figure 1 shows how d is represented in each step of the phases (a) and (b). The original sentence (1) is tokenized and, for each token, part of speech ambiguities are

[1] www.imdb.com

solved (2). Reduction to lemmas (3)(for example, verbs are turned to their base form) is performed before deleting stopwords (4). Then, each word is assigned to the most appropriate sense, represented by a sense identifier obtained from WordNet(5).

```
The     white      cat     is      hunting      the     mouse     (1)
The/DT white/JJ  cat/NN is/VBZ hunting/VBG the/DT mouse/NN (2)
The/DT white/JJ  cat/NN be/VB  hunt/VB      the/DT mouse/NN (3)
       white/JJ  cat/NN         hunt/VB              mouse/NN (4)
       00373636  02037721       01108575             02244530 (5)
```

Fig. 1. The preprocessing of sentence "The white cat is hunting the mouse". Each token is labeled with a tag describing its lexical role in the sentence. NN=noun, singular - VB=verb, base form - VBZ=verb, is - VBG=verb, gerund form - JJ=adjective, DT=determinative. According to its role, each token is assigned to the most appropriate sense.

As for lemmatization and part-of-speech tagging we use the MontyLingua natural language processor[2] for English. Document d, after step (4) in Figure 1, is the input for the synset identification phase. The core idea behind the proposed WSD algorithm is to disambiguate w by determining the degree of *semantic similarity* among candidate synsets for w and those of each word in C. Thus, the proper synset assigned to w is that with the highest similarity with respect to its context of use. A crucial point is the choice of a suitable similarity measure, by taking into account the specialness of the user profiling task we are addressing. In the following, we discuss the choice of the semantic similarity adopted in the WSD algorithm, before describing the complete procedure.

The semantic similarity measure. A natural way to evaluate semantic similarity in a taxonomy is to evaluate the distance between the nodes corresponding to the items being compared. The shorter the path from one node to another, the more similar they are. The measure of semantic similarity adopted in this work is the Leacock-Chodorow measure [9], which is based on the length of the path between concepts in an IS-A hierarchy. The idea behind this measure is that similarity between synsets a and b is inversely proportional to the distance between them in the WordNet *is-a* hierarchy, measured by the number of nodes in the shortest path (the path having minimum number of nodes) from a to b . The similarity is computed in the proposed WSD algorithm by the function `SinSim` (lines 24-28): the path length N_p is scaled by the depth D of the hierarchy, where depth is defined as the length of the longest path from a leaf node to the root node of the hierarchy. In a study conducted by [14], it is performed a detailed analysis of the performances of several similarity measures using a variety of different sources to determine the semantic relatedness of words. The main finding of the study is that measures combining the structure of WordNet with information content values taken from corpora provided better results

[2] http://web.media.mit.edu/hugo/montylingua

with respect to measures that rely only on the concept hierarchy structure or information content values. Information content of a concept is a measure of the specificity of a concept in a hierarchy. It is usually estimated by counting the frequency of that concept in a *large* corpus. If sense-tagged text is available, frequency counts of concepts can be attained directly, since each concept will be associated with a unique sense. If sense tagged text is not available (which is the usual situation), it will be necessary to adopt an alternative counting scheme. For example, Resnik [16] suggests counting the number of occurrences of a word in a corpus, and then dividing that count by the number of different senses associated with that word. This value is then assigned to each concept. In our case, disambiguation is performed for the specific task of building a user profile. Therefore, the corpus that should be adopted to estimate the frequency of concepts is the set of documents on which the user provided ratings. It is unreasonable to assume that this corpus is annotated with senses or that it is sufficiently large to perform an alternative counting scheme as the one suggested by Resnik. These problems do not allow to adopt measures based on corpus frequencies and lead us to rely on an approach exclusively based on the knowledge coming from WordNet.

The Word Sense Disambiguation procedure. In this section we describe the WSD procedure based on the Leacock-Chodorow measure, and analyze each step by using the sentence *"The white cat is hunting the mouse"* as example. Let w="cat" be the word to be disambiguated. The procedure starts by defining the context C of w as the set of words in the same slot of w having the same POS as w. In this case, the only *noun* in the sentence is "mouse", then $C = \{mouse\}$. Next, the algorithm identifies both the sense inventory for w, that is `X = {01789046: feline mammal, 00683044: computerized axial tomography,...}`, and the sense inventory X_j for each word w_j in C. Thus, X_j= `{01993048: small rodents, 03304722: a hand-operated electronic device that controls the coordinates of a cursor, ... }`. The sense inventory T for the whole context C is given by the union of all X_j (in this case, as C has a single word, then $X_j = T$). After this step, we measure the similarity of each candidate sense $s_i \in X$ to that of each sense $s_h \in T$ and then the sense assigned to w is the one with the highest similarity score. In the example, `SinSim(01789046: feline mammal, 01993048: small rodents) = 0.806` is the highest similarity score, thus w is interpreted as "feline mammal". Each document is mapped into a list of WordNet synsets following three steps:

1. each monosemous word w in a slot of a document d is mapped into the corresponding WordNet synset;
2. for each pair of words $\langle$*noun,noun*$\rangle$ or $\langle$*adjective,noun*$\rangle$, a search in WordNet is made to verify if at least one synset exists for the bigram $\langle w_1, w_2 \rangle$. In the positive case, algorithm 1 is applied on the bigram, otherwise it is applied separately on w_1 and w_2; in both cases all words in the slot are used as the context C of the word(s) to be disambiguated;

Algorithm 1 The WordNet-based WSD algorithm

```
1: procedure WSD(w, d)        ▷ finds the proper synset of a polysemous word w in document d
2:     C ← {w_1, ..., w_n}        ▷ C is the context of w. For example, C = {w_1, w_2, w_3, w_4} is a window with radius=2, if the sequence of words {w_1, w_2, w, w_3, w_4} appears in d
3:     X ← {s_1, ...s_k}    ▷ X is sense inventory for w, that is the set of all candidate synsets for w returned by WordNet
4:     s ← null        ▷ s is the synset to be returned
5:     score ← 0    ▷ score is the similarity score assigned to s wrt to the context C
6:     T ← ∅        ▷ T is the set of all candidate synsets for all words in C
7:     for all w_j ∈ C do
8:         if POS(w_j) = POS(w) then        ▷ POS(y) is the part-of-speech of y
9:             X_j ← {s_j1, ...s_jm}        ▷ X_j is the set of m possible senses for w_j
10:            T ← T ∪ X_j
11:        end if
12:    end for
13:    for all s_i ∈ X do
14:        for all s_h ∈ T do
15:            score_ih ← SinSim(s_i, s_h)    ▷ computing similarity scores between s_i and every synset s_h ∈ T
16:            if score_ih ≥ score then
17:                score ← score_ih
18:                s ← s_i  ▷ s is the synset s_i ∈ X having the highest similarity score wrt the synsets in T
19:            end if
20:        end for
21:    end for
22:    return s
23: end procedure
24: function SinSim(a, b)        ▷ The similarity of the synsets a and b
25:    N_p ←the number of nodes in path p from a to b
26:    D ←maximum depth of the taxonomy        ▷ In WordNet 1.7.1 D = 16
27:    r ← −log(N_p/2 · D)
28:    return r
29: end function
```

3. each polysemous unigram w is disambiguated by algorithm 1, using all words in the slot as the context C of w.

Our hypothesis is that the proposed indexing procedure helps to obtain profiles able to recommend documents semantically closer to the user interests. The difference with respect to keyword-based profiles is that synset unique identifiers are used instead of words. As an example, Figure 3 shows a fragment of the BOS representation for the document presented in Figure 2. For readability reasons, we show the natural language description of the synset provided by WordNet, in addition to the synset unique identifier used in the actual implementation and the number of occurrences of the synset.

```
title:  The Shining
director: Stanley Kubrick
cast:   Jack Nicholson, Shelley Duvall, Danny Lloyd,
Scatman Crothers, Barry Nelson, Philip Stone,
Joe Turkel, Anne Jackson, Tony Burton, Lia
Beldam, Billie Gibson, Barry Dennen...
summary:  A male novelist is having writer's block.
He, his wife, and his young son become
the care-takers of a haunted hotel so
he can go back to writing again. Once they
start meeting the ghosts, they talk to
them by ''shining'' (telepathic conversation)...
keywords: extrasensory-perception, freeze-to-death, bar,
  axe-murder, psychological-drama, child-in-peril,
  whiskey, murder, winter...
```

Fig. 2. The five slots corresponding to the description of the movie "The Shining"

```
title: {shining-the work of making something shine by polishing it;
"the shining of shoes provided a meager living"-
434048: 1.0}
director: {stanley kubrick-United States filmmaker (born in 1928)-
9111534: 1.0}
cast: {}
summary: {male-(biology) being the sex (of plant or animal) that
  produces gametes (spermatozoa) that perform the
  fertilizing function in generation; "a male infant";
  "a male holly tree"-1432909: 1.0,
  novelist-someone who writes novels-
  8492863: 2.0,...

keywords: {extrasensory perception-apparent power to perceive
 things that are not present to the senses-
 6047688: 1.0, freeze-be cold-00075821: 1.0,
 death-the event of dying or departure from life-06904072: 1.0;
 ...}
```

Fig. 3. The Bag-of-synsets representation of the movie "The Shining"

4 A Relevance Feedback Method for Learning WordNet-Based Profiles

In the Rocchio algorithm, documents are represented with the vector space model and the major heuristic component is the TFIDF word weighting scheme [17]:

$$\text{TFIDF}(t_k, d_j) = \underbrace{\text{TF}(t_k, d_j)}_{\text{TF}} \cdot \underbrace{log\frac{N}{n_k}}_{\text{IDF}} \tag{1}$$

where N is the total number of documents in the training set and n_k is the number of documents containing the term t_k. $TF(t_k, d_j)$ computes the frequency of t_k in document d_j. Learning combines vectors of positive and negative examples into a prototype vector $\overrightarrow{c}$ for each class in the set of classes C. The method computes a classifier $\overrightarrow{c_i} = \langle \omega_{1i}, \ldots, \omega_{|T|i} \rangle$ for category c_i (T is the *vocabulary*, that is the set of distinct terms in the training set) by means of the formula:

$$\omega_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{\omega_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{\omega_{kj}}{|NEG_i|} \tag{2}$$

where ω_{kj} is the TFIDF weight of the term t_k in document d_j, POS_i and NEG_i are the set of positive and negative examples in the training set for the specific class c_i, β and γ are control parameters that allow setting the relative importance of *all* positive and negative examples. To assign a class $\widetilde{c}$ to a document d_j, the similarity between each prototype vector $\overrightarrow{c_i}$ and the document vector $\overrightarrow{d_j}$ is computed and $\widetilde{c}$ will be the c_i with the highest value of similarity. We propose a modified version of this method able to manage documents structured in slots and represented by WordNet synsets. As reported in Section 2.1, each document d_j is represented in the vector space by five synset-frequency vectors:

$$f_j^m = \langle w_{j1}^m, w_{j2}^m, \ldots, w_{jD_{jm}}^m \rangle$$

where D_{jm} is the total number of different synsets appearing in the m-th slot of document d_j and w_{jk}^m is the weight of the synset t_k in the slot s_m of document d_j, computed according to a synset weighting strategy described in the next section.

4.1 Synset Weighting Strategy

Term selection techniques scores each term in T, the set of all terms in the training set, by a class-based Term Evaluation Function (TEF) f, and then selects a set T' of terms that maximize f. TEFs used in TC try to capture the intuition according to which the most valuable terms for categorization under c_i are those that are distributed most differently in the sets of positive and negative examples of c_i [20]. In [5], it is proposed that TEFs could be better substitutes of IDF-like functions. Instead of discarding scores that TEFs attribute to terms after selecting those that will be included in the document representation, they are used also in the term weighting phase. According to this idea, we propose the use of *Synset Evaluation Functions* (SEFs) in the synset weighting phase. The proposed SEFs are obtained by modifying two TEFs: the *Gain Ratio* [20] and the *Max Term Frequency-Square Page Frequency* [4]. The modified *Gain Ratio* computes how much information the synset t_k in slot s_m gives about class c_i:

$$GR(t_k, c_i, s_m) = \frac{\sum_{c \in \{c_i, \overline{c_i}\}} \sum_{t \in \{t_k, \overline{t_k}\}} P(t, c, s_m) log_2 \frac{P(t,c,s_m)}{P(t,s_m)P(c)}}{-\sum_{c \in \{c_i, \overline{c_i}\}} P(c) log_2 P(c)} \tag{3}$$

The score of a synset t_k that appears in the slot s_m of a document d_j belonging to class c_i is computed as:

$$w_{kj}^m = \text{SFIDF}(t_k, d_j, s_m) \cdot SEF(t_k, c_i, s_m) \tag{4}$$

where $\text{SFIDF}(t_k, d_j, s_m)$ is the synset frequency-inverse document frequency, computed as in Equation (5) by counting occurrences of the synsets separately in each slot. $SEF(t_k, c_i, s_m)$ is the score computed by the selected synset evaluation function. Notice that, in our profile learning problem, item descriptions belong to specific categories: this means that we consider movies already classified by "genre" (*horror*, *action*, etc.). Our aim is to learn a profile of preferred movies by a user for each "genre" G he/she provided ratings. This condition is important when computing $\text{SFIDF}(t_k, d_j, s_m)$:

$$\text{SFIDF}(t_k, d_j, s_m) = \text{SF}(t_k, d_j, s_m) \cdot \underbrace{log \frac{|G|}{\#G(t_k, s_m)}}_{\text{IDF}} \tag{5}$$

where $|G|$ is the number of documents in genre G, $\#G(t_k, s_m)$ denotes the number of documents in "genre" G in which t_k occurs at least once in slot s_m. $\text{SF}(t_k, d_j, s_m)$ is computed as follows:

$$\text{SF}(t_k, d_j, s_m) = \begin{cases} 1 + log\#(t_k, d_j, s_m) & \text{if } \#(t_k, d_j, s_m) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

In Equation (6) $\#(t_k, d_j, s_m)$ denotes the number of times t_k occurs in slot s_m of document d_j. The idea in Equation (4) is that the most informative synsets of user preferences for genre G are rare synsets for G (high IDF value) that are distributed most differently in the sets of positive and negative examples of c_i (high SEF value). Thus, we decided to use also the IDF score in our weighting approach, instead of replacing it by the SEF scores, as suggested in [5]. The other SEF we employ in our study is obtained by modifying the TEF presented in [4], where authors show that the proposed feature selection technique compares favorably with respect to other well-known approaches. However, we modified this measure to adjust it to the slot document representation. Given the training document d_j, belonging to class c_i, for each synset t_k in the slot s_m, the frequency $\text{SF}(t_k, d_j, s_m)$ of the synset in the document is computed. Then, for each class c_i, synset t_k, and slot s_m, the following statistics are computed:

- $\text{MaxSF}(t_k, c_i, s_m)$, the maximum value of $\text{SF}(t_k, d, s_m)$ on all training documents d of class c_i;
- $DF(t_k, c_i, s_m)$, the document frequency, that is, the percentage of documents of class c_i in which the synset t_k occurs in the slot s_m;
- $ICF(t_k, s_m) = 1/CF(t_k, s_m)$, where $CF(t_k, s_m)$ (class frequency) is the number of class in which the synset t_k occurs in slot s_m.

The score $SEF(t_k, c_i, s_m)$ is given by the product of MaxSF, DF and ICF. We call this measure *Max Synset Frequency-Document Frequency*. We introduced

also another variant of MaxSF-DF-ICF that takes into account both document representation and ratings given by users. This measure, that we call *Weighted Max Synset Frequency-Document Frequency* (weighted MaxSF-DF-ICF), uses ratings given by users to weight the occurrences of synsets and to compute DF and ICF (weights range between 0 and 1). The statistics are modified as follows:

- $\text{MaxSF}(t_k, c_i, s_m)$ - the *weighted* maximum value of $\text{SF}(t_k, d, s_m)$ on all training documents d of class c_i, where *occurrences are weighted using ratings*. For example, if the maximum number of occurrences of t_k in the slot s_m of documents in class c_i is 5, and, given that the weight of d_j (the document in which the maximum number of occurrences is observed) in c_i is 0.7, then $\text{MaxSF}(t_k, c_i, s_m) = 3.5$;
- $DF(t_k, c_i, s_m)$ - the *weighted* document frequency, that is, the *weighted* percentage of documents of class c_i in which the synset t_k occurs in the slot s_m. For example, consider d_1 (weight=1.0) and d_2 (weight=0.6) belonging to c_i. If t_k occurs in slot s_m of d_1, then $DF(t_k, c_i, s_m) = 1.0/1.6 = 0.625$, while in the *not-weighted* variant $DF(t_k, c_i, s_m) = 0.5$.
- $ICF(t_k, c_i, s_m)$ - the *weighted* inverse category frequency, computed as:

$$ICF(t_k, c_i, s_m) = \frac{1}{1 + \sum_{j \neq i} DF(t_k, c_j, s_m)} \tag{7}$$

  For example, let's consider d_1 (weight=0.8) and d_2 (weight=0.6), belonging to class c_+, and d_3 (weight=0.2) and d_4 (weight=0.4), belonging to class c_-. If t_k occurs in slot s_m both of d_1 and d_3, then $ICF(t_k, c_+, s_m) = 0.75$ and $ICF(t_k, c_-, s_m) = 0.636$, while in the *not-weighted* variant $ICF(t_k, s_m) = 0.5$. In the *not-weighted* variant, the ICF score is the same for all classes, because we don't consider the weights of the documents in which t_k appears. In the *weighted* variant, if a synset appears in both classes, we take into account if documents belonging to one class in which t_k occurs are "heavier" than documents belonging to the other class in which t_k appears.

The final SEF score is computed as for *not-weighted* variant. In conclusion, in the experiments reported in section 5, we use three different SEFs: 1) *Gain Ratio*, Equation (3); 2) MaxSF-DF-ICF; 3) *weighted* MaxSF-DF-ICF.

4.2 Synset-Based Profiles

Given a user u and a set of rated movies in a specific genre (e.g. *Comedy*), the aim is to learn a profile able to recognize movies liked by the user in that genre. Learning consists in inducing one prototype vector for *each slot*: these five vectors will represent the user profile. Each prototype vector could contribute in a different way to the calculation of the similarity between the vectors representing a movie and the vectors representing the user profile. The algorithm learns two different profiles $\overrightarrow{p_i} = \langle \omega^m_{1i}, \ldots, \omega^m_{|T_m|i} \rangle$, for a user u and a category c_i by using the ratings given by the user on documents in c_i. The rating $r_{u,j}$ on the document d_j is a discrete judgment ranging from 1 to 6 used to compute the coordinates of the vectors in both the positive and the negative user profile:

$$\omega_{ki}^{m} = \sum_{\{d_j \in POS_i\}} \frac{\omega_{kj}^{m} \cdot r'_{u,j}}{|POS_i|} \quad (8) \qquad \omega_{ki}^{m} = \sum_{\{d_j \in NEG_i\}} \frac{\omega_{kj}^{m} \cdot r'_{u,j}}{|NEG_i|} \quad (9)$$

where $r'_{u,j}$ is the normalized value of $r_{u,j}$ ranging between 0 and 1 (respectively corresponding to $r_{u,j} = 1$ and 6), $POS_i = \{d_j \in T_r | r_{u,j} > 3\}$, $NEG_i = \{d_j \in T_r | r_{u,j} \leq 3\}$, and ω_{kj}^{m} is the weight of the synset t_k in the slot s_m of document d_j, computed as in equation (4), where the IDF factor is computed over POS_i or NEG_i depending on the fact that the synset t_k is in the slot s_m of a movie rated as positive or negative (if the synset is present in both positive and negative movies two different values for it will be computed). Computing two different IDF values for a synset led us to consider the rarity of a synset in positive and negative movies, in an attempt to catch the informative power of a synset in recognizing interesting movies. Equations (8) and (9) differ from the classical formula in the fact that the parameters β and γ are substituted by the ratings $r'_{u,j}$ that give a different weight to each document in the training set. The similarity between a profile $\overrightarrow{p_i}$ and a movie $\overrightarrow{d_j}$ is obtained by computing five partial similarity values between each pair of corresponding vectors in $\overrightarrow{p_i}$ and $\overrightarrow{d_j}$. A weighted average of the five values is computed, assigning a different weight α_s to reflect the importance of a slot in classifying a movie. In our experiments, we used $\alpha_1 = 0.1$ (title), $\alpha_2 = 0.15$ (director), $\alpha_3 = 0.15$ (cast), $\alpha_4 = 0.25$ (summary) and $\alpha_5 = 0.35$ (keywords). The values α_s were decided according to experiments not reported in the paper due to space limitations. We considered different values for each α_s and repeated the experiments reported in section 5 using the selected values. The values reported here are those that allowed to obtain the best predictive accuracy. Since the user profile is composed by both the positive and the negative profiles, we compute two similarity values, one for each profile. The document d_j is considered as interesting only if the similarity value of the positive profile is higher than the similarity of the negative one.

5 Experimental Sessions

The goal of experiments was to evaluate if synset-based profiles had a better performance than word-based profiles. Experiments were carried out on a collection of 1,628 textual descriptions of movies rated by 72,916 real users, the EachMovie dataset. Movies are rated on a 6-point scale mapped linearly to the interval [0,1]. The content of each movie was collected from the Internet Movie Database[3] by a crawler. Tokenization, stopword elimination and stemming have been applied to obtain the BOW. Documents indexed by the BOS model have been processed by tokenization, stopword elimination, lemmatization and WSD. Movies are categorized into different genres. For each genre or category, a set of 100 users was randomly selected among users that rated n items, $30 \leq n \leq 100$ in that movie category (only for genre 'animation', the number of users that rated n movies was 33, due to the low number of movies if that genre). For each

[3] IMDb, http://www.imdb.com

Table 1. 10 'Genre' datasets obtained from the original EachMovie dataset

Id Genre	Genre	Number of Movies rated	% POS	% NEG
1	Action	4,474	72	28
2	Animation	1,103	57	43
3	Art_Foreign	4,246	76	24
4	Classic	5,026	92	8
5	Comedy	4,714	63	37
6	Drama	4,880	76	24
7	Family	3,808	64	36
8	Horror	3,631	60	40
9	Romance	3,707	73	27
10	Thriller	3,709	72	28
		39,298	72	28

category, a dataset of at least 3000 triples (user,movie,rating) was obtained (at least 990 for 'animation'). Table 1 summarizes the data used for experiments. The number of movies rated as positive and negative in that genre is balanced in datasets 2, 5, 7, 8 (55-70 % positive, 30-45% negative), while is unbalanced in datasets 1, 3, 4, 6, 9, 10 (over 70% positive). Documents have been disambiguated using Algorithm 1, obtaining a feature reduction of 38% (172, 296 words vs. 107, 990 synsets). This is mainly due to the fact that bigrams are represented using only one synset and that synonym words are represented by the same synset. Classification effectiveness was evaluated by the classical measures *precision*, *recall* and *F-measure* [20]. We adopted the Normalized Distance-based Performance Measure (NDPM) [22] to measure the distance between the ranking imposed on items by the user ratings and the ranking predicted by the Rocchio method, that ranks items according to the similarity to the profile of the class *likes*. Values range from 0 (agreement) to 1 (disagreement). In the experiments, a movie is considered as *relevant* by a user if the rating $r \geq 3$, while the Rocchio method considers an item as relevant if the similarity score for the class *likes* is higher than the one for the class *dislikes*. We executed one experiment for each user. Each experiment consisted in 1) selecting ratings of the user and the content of the movies rated by that user; 2) splitting the selected data into a training set Tr and a test set Ts; 3) using Tr for learning the corresponding user profile; 4) evaluating the predictive accuracy of the induced profile on Ts, using the aforementioned measures. The methodology adopted for obtaining Tr and Ts was the 10-fold cross validation. Table 2 defines the experimental plan and reports results obtained on average over all 10 genres. Results subdivided by genre are reported in Table 3 only for experiment D, that provided the better performance. The first column of the Table 2 indicates the experiment identifier, the second one defines whether user ratings or simply a binary relevance judgment have been used to label training examples. In case of binary feedback, parameters used to weight positive and negative training examples were respectively $\beta = 16$ and $\gamma = 4$ (see [20] for more details). The third column specifies the Synset Evaluation Functions used in the experiments.

Table 2. Experimental plan and performance of profiles in the two models

Exp	Ratings	Synset Evaluation Function	Precision		Recall		F1		NDPM	
			BOW	BOS	BOW	BOS	BOW	BOS	BOW	BOS
A	Y	N	0.74	0.75	0.80	0.82	0.75	0.77	0.45	0.44
B	Y	Gain Ratio	0.73	0.74	0.80	0.83	0.75	0.77	0.43	0.44
C	Y	MaxSF-DF-ICF	0.74	0.76	0.78	0.80	0.74	0.76	0.45	0.44
D	Y	Weighted MaxSF-DF-ICF	0.74	0.76	0.81	0.84	0.76	0.78	0.44	0.44
E	N	N	0.73	0.76	0.79	0.83	0.75	0.77	0.45	0.45
F	N	Gain Ratio	0.70	0.74	0.76	0.83	0.71	0.77	0.43	0.44
G	N	MaxSF-DF-ICF	0.74	0.76	0.76	0.79	0.73	0.75	0.45	0.45
Mean			0.73	0.75	0.79	0.82	0.74	0.77	0.44	0.44

Table 3. Comparison between the BOW and the BOS approach

Id Genre	Precision		Recall		F1		NDPM	
	BOW	BOS	BOW	BOS	BOW	BOS	BOW	BOS
1	0.74	0.75	0.84	0.86	0.76	0.79	0.46	0.44
2	0.65	0.64	0.70	0.70	0.68	0.63	0.34	0.38
3	0.77	0.85	0.80	0.87	0.77	0.84	0.46	0.48
4	0.92	0.94	0.94	0.96	0.93	0.94	0.45	0.43
5	0.67	0.69	0.72	0.75	0.67	0.70	0.44	0.46
6	0.78	0.79	0.84	0.87	0.80	0.81	0.45	0.45
7	0.68	0.74	0.79	0.84	0.73	0.77	0.41	0.40
8	0.64	0.69	0.78	0.84	0.69	0.73	0.42	0.44
9	0.75	0.76	0.83	0.85	0.76	0.77	0.48	0.48
10	0.74	0.75	0.84	0.85	0.77	0.78	0.45	0.44
Mean	0.74	0.76	0.81	0.84	0.76	0.78	0.44	0.44

In the BOW model, evaluation functions are used to weight words instead of synsets.

From the results in Table 2, we can say that the use of ratings as parameters to weigh training examples produces a positive effect on the performance of the classifier, with both the BOW-based indexing and the BOS-based one. By observing the behavior of the different synset evaluation functions, we note that the best performance is obtained in experiment D, by using the proposed Weighted Max Synset Frequency-Document Frequency. Specifically, both precision and recall increased (+2% and +3%, respectively), thus also F-measure improved. In more detail (Table 3), the BOS model outperforms the BOW model in precision on datasets 3 (+8%), 7 (+6%), and 8 (+5%). No improvement has been observed only on dataset 2. This is probably due both to the low number of ratings and to the specific features of the movies, in most cases stories, that make difficult the disambiguation task. Similar results have been observed as regards recall and F-measure. This could be an indication that the improved results are independent from the distribution of positive and negative examples in the datasets: the number of movies rated as positive and negative is balanced

in datasets 7 and 8, while is strongly unbalanced in dataset 3. NDPM has not been improved, but it remains acceptable. This measure compared the ranking imposed by the user ratings and the similarity score for the class c_+: further investigations will be carried out to define a better ranking score for computing NDPM, that will take into account the negative part of the profile as well. It could be noticed from the NDPM values that the relevant / not relevant classification is improved without improving the ranking. The general conclusion is that the BOS method has improved the classification of items whose score (and ratings) is close to the relevant / not relevant threshold, thus items for which the classification is highly uncertain (thus minor changes in the ranking have not modified the NDPM values). A Wilcoxon signed ranked test ($p < 0.05$) has been performed to validate the results. We considered each experiment as a single trial for the test. The test confirmed that there is a statistically significant difference in favor of the BOS model with respect to the BOW model as regards precision, recall and F-measure, and that the two models are equivalent in defining the ranking of the preferred movies with respect to the score for the class "likes".

6 Conclusions and Future Work

We presented a framework for content-based retrieval integrating a relevance feedback method with a WSD strategy based on WordNet for inducing semantic user profiles. Our hypothesis is that substituting words with synsets produces a more accurate document representation that could be successfully used by learning algorithms to infer more accurate user profiles. This hypothesis is confirmed by the experimental results, since, as expected, a synset-based classification allows to prefer documents with high degree of semantic coherence, which is not guaranteed in case of a word-based classification. As a future work, we will evaluate the effectiveness of the WSD algorithm, by comparing its performance to state-of-the-art systems. Moreover, a comparison of BOS representation with other techniques that replace words by "topics" [6] will be carried out.

Acknowledgments

This research was partially funded by the European Commission under the 6th Framework Programme IST Integrated Project VIKEF No. 507173, Priority 2.3.1.7 Semantic-based Knowledge Systems - `http://www.vikef.net`.

References

1. F. Asnicar and C. Tasso. ifweb: a prototype of user model-based intelligent agent for documentation filtering and navigation in the word wide web. In *Proc. of 1st Int. Workshop on adaptive systems and user modeling on the WWW*, 1997.
2. M. Balabanovic and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

3. S. Bloedhorn and A. Hotho. Boosting for text classification with semantic features. In *Proc. of 10th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining, Mining for and from the Semantic Web Workshop*, pages 70–87, 2004.
4. M. Ceci and D. Malerba. Hierarchical classification of HTML documents with WebClassII. In F. Sebastiani, editor, *Proceedings of ECIR-03, 25th European Conference on Information Retrieval*, pages 57–72, Pisa, IT, 2003. Springer Verlag.
5. F. Debole and F. Sebastiani. Supervised term weighting for automated text categorization. In *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing*, pages 784–788, Melbourne, US, 2003. ACM Press, New York, US.
6. S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'88*, 1988.
7. C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
8. A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *Proc. of the SIGIR Semantic Web Workshop*, 2003.
9. C. Leacock and M. Chodorow. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press, 1998.
10. B. Magnini and C. Strapparava. Improving user modelling with content-based techniques. In *Proc. 8th Int. Conf. User Modeling*, pages 74–83. Springer, 2001.
11. C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, US, 1984.
12. D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, and G. Weikum. Word sense disambiguation for exploiting hierarchical thesauri in text classification. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 181–192. Springer, 2005.
13. D. Mladenic. Text-learning and related intelligent agents: a survey. *IEEE Intelligent Systems*, 14(4):44–54, 1999.
14. S. Patwardhan, S. Banerjee, and T. Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proc. of the Fourth Intern. Conf. on Intelligent Text Processing and Computational Linguistics*, page 241, 2003.
15. M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.
16. P. Resnik. *WordNet and class-based probabilities*, pages 305–332. In C. Fellbaum (Ed.), MIT Press, 1998.
17. J. Rocchio. Relevance feedback information retrieval. In G. Salton, editor, *The SMART retrieval system - experiments in automated document processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.
18. M. d. B. Rodriguez, J. M. Gomez-Hidalgo, and B. Diaz-Agudo. Using wordnet to complement training information in text categorization. In *2nd Int. Conf. on Recent Advances in NLP*, pages 150–157, 1997.
19. S. Scott and S. Matwin. Text classification using wordnet hypernyms. In *COLING-ACL Workshop on usage of WordNet for in NLP Systems*, pages 45–51, 1998.
20. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 2002.
21. M. Theobald, R. Schenkel, and G. Weikum. Exploting structure, annotation, and ontological knowledge for automatic classification of xml data. In *Proceedings of International Workshop on Web and Databases*, pages 1–6, 2004.
22. Y. Y. Yao. Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information Science*, 46(2):133–145, 1995.

Visibility Analysis on the Web Using Co-visibilities and Semantic Networks

Peter Kiefer, Klaus Stein, and Christoph Schlieder

Laboratory for Semantic Information Processing
Otto-Friedrich-University Bamberg, Germany
{peter.kiefer, klaus.stein, christoph.schlieder}@wiai.uni-bamberg.de

Abstract. Monitoring public attention for a topic is of interest for many target groups like social scientists or public relations. Several examples demonstrate how public attention caused by real-world events is accompanied by an accordant visibility of topics on the web. It is shown that the hitcount values of a search engine we use as initial visibility values have to be adjusted by taking the semantic relations between topics into account. We model these relations using semantic networks and present an algorithm based on Spreading Activation that adjusts the initial visibilities. The concept of co-visibility between topics is integrated to obtain an algorithm that mostly complies with an intuitive view on visibilities. The reliability of search engine hitcounts is discussed.

1 Introduction

Social scientists have invested much effort in manually analyzing daily news while trying to monitor public awareness for certain topics (see e. g. [1]). Especially in nowadays information society, the topics that are visible in public discussions across different kinds of media tend to change rapidly. It becomes increasingly important for organizations to be present in the minds of people and to evaluate public relations activities [2], be it a company competing for customers' attention or a non-profit organization trying to arouse public awareness for their concerns (see also work on attention economies, e. g. [3]). The undoubted primacy of the internet raises the question whether public visibility of topics goes along with an accordant visibility of these topics on the web. If such a correlation between real world events and online visibility exists, monitoring topics on the web could give an important indicator for the target groups mentioned above.

In this paper, we aim at providing methods to support the monitoring of the visibility of topics on the internet. We do not deal with topic detection, but assume a user who previously knows the topics of interest. We thereby take a quite broad view of what is regarded as a topic: anything that can draw public attention on itself (and is expressible by some kind of search term), ranging from typical discussion group topics like 'climate policy' to persons like 'George Bush' or even something basic like 'Christmas'. We propose a simple way to measure the visibility of topics, based on hitcount values of a search engine,

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 34–50, 2006.

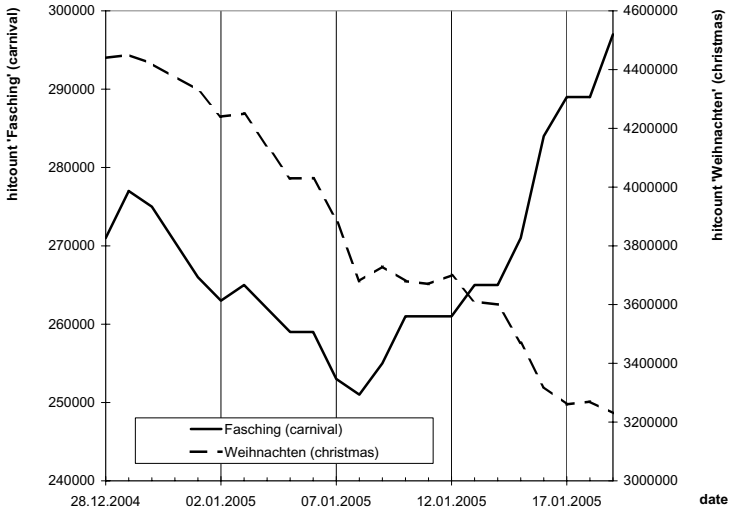


Fig. 1. Estimated hitcount values (Google) for 'Weihnachten' (Christmas) and 'Fasching' (Carnival) in time

present examples indicating that real world events actually do have an impact on visibility on the web and introduce the concept of topic co-visibility (section 2).

It is often not sufficient to monitor just a single topic, rather several semantically related topics need to be observed simultaneously. We show how to correct our initial visibility values by adding knowledge about the semantical relations between topics (section 3). In this context, we contribute a new algorithm based on Spreading Activation (section 4).

In section 5 we report on the experiences we made concerning the reliability of the hitcounts of the search engine we used for our case studies. At last (section 6) we give a short summary and a view on current research.

2 Visibility and Co-visibility

2.1 Visibility

Our first objective was to find an appropriate measure for the visibility of a topic in internet communication processes. However, possible measures depend on the communication process analyzed, for instance messages in a newsgroup should be treated differently than a collection of documents without link structure. We define the visibility of a given topic by vis(*top*) = hitcount("top") with hitcount("top") being the number of pages found on the search term "top" by a given search engine.[1]

[1] For all examples given in this paper we used the estimated hitcount values of the Google Web API (`http://www.google.com/apis/`). Note that hitcount values from search engines (especially from Google) are usually estimated and not exact.

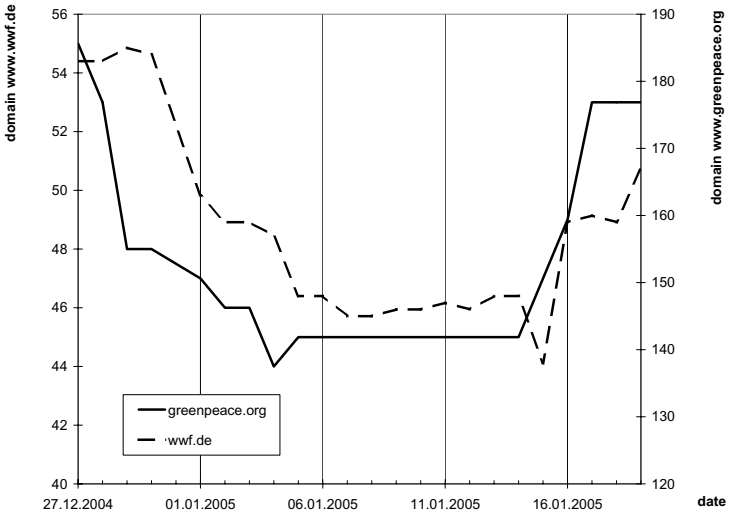


Fig. 2. Estimated hitcount values (Google) for 'Klimapolitik' (climate policy) on the domains `www.greenpeace.org` and `www.wwf.de` in time

Fig. 1 shows the developing of the visibility for the topics 'Weihnachten'[2] (Christmas) and 'Fasching' (Carnival) from Dec. 28, 2004 to Jan. 19, 2005. Obviously, the course of seasons leaves its traces on the internet. The visibility of 'Weihnachten' actually decreased by 25%. This is not a trivial finding for often web pages are created for a certain event but not necessarily removed afterwards, so we did not anticipate such a rapid decrease. The continuous growth of the web suggested that most of the webpages are kept.

The simultaneous change of visibility of one topic in different places is shown in Fig. 2, monitoring the topic 'Klimapolitik' (climate policy) from Dec. 27, 2004 to Jan. 19, 2005 in the two domains `www.greenpeace.org` and `www.wwf.de`. This clearly demonstrates the similarity of discussed topics among different sources. We will return to the example of climate policy below.

An impressive use case for the usability of the simple hitcount visibility measure in the context of marketing evaluation is described in [4] and should be mentioned at this point: in January 2005, a German company from the pharmaceutical branch (Dr.Kurt Wolff GmbH&Co. KG, brand name Alpecin) launched a new hair liquid called 'After Shampoo Liquid' with a special chemical compound as new ingredient, the 'Coffein-Complex'. There were marketing attempts in German media to promote this 'After Shampoo Liquid'. Commercials were emphasizing the 'Coffein-Complex' and encouraging consumers to visit the company's website and try the 'Glatzenrechner' ('balding calculator'[3]). A successful marketing campaign should draw public attention on the product and therefore

[2] All analysis for this paper was done in German.
[3] http://www.alpecin.de/en/balding-calculator/

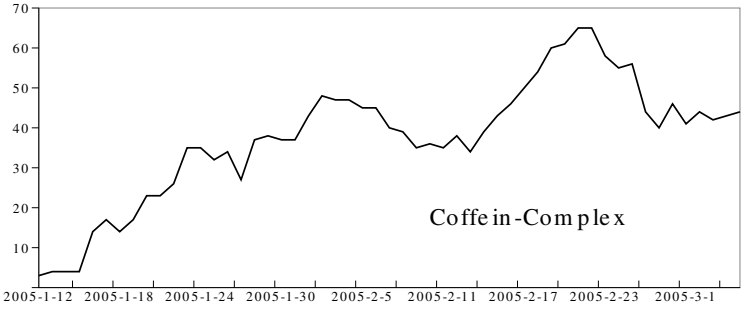


Fig. 3. Estimated hitcount values (Google) for 'Coffein-Complex' in time

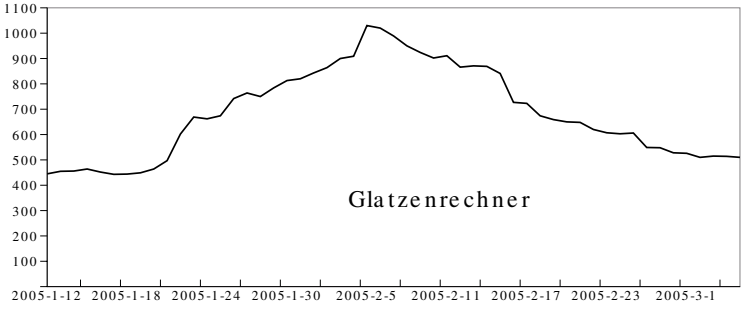


Fig. 4. Estimated hitcount values (Google) for 'Glatzenrechner' (balding calculator) in time

raise public visibility. We monitored the topics 'Coffein-Complex' (Fig. 3) and 'Glatzenrechner' (Fig. 4) from Jan. 12, 2005 to Mar. 5, 2005 and detected significant changes in visibility: 'Coffein-Complex' started with a hitcount of 3 and increased up to 65 on Feb. 22 before going down to the level of around 43. This shows how a product-related term or technology that did almost not exist on the internet can gain visibility through marketing actions. 'Glatzenrechner' was already present with a hitcount of 445, but more than doubled its hitcount to reach a maximum of 1030 on Feb. 5 before it approached a hitcount around 500.

Although the idea to measure visibility by hitcount values seems trivial and does not take the link structure or additional information into account, it has three main advantages:

1. It is based on existing search engines and therefore implemented quite easily.
2. It allows automated daily monitoring with only little effort.
3. It scales from monitoring visibilities from a certain domain to the whole (accessible) internet.

Defining topic visibility by the hitcount of one search term will hardly suit all use cases. Complex topics like 'US foreign policy during the cold war and its impacts on the German economy' often do not fulfill this requirement. However,

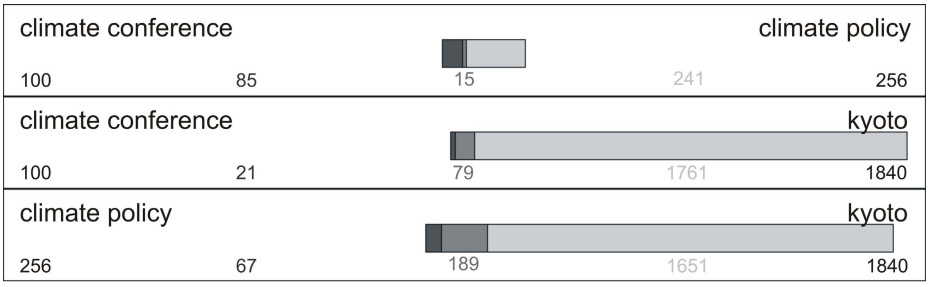


Fig. 5. Bar visualization of hitcount- and co-hitcount values (Google) for 'climate policy', 'climate conference' and 'Kyoto' (on `www.greenpeace.org`) at July 23, 2005. The graph shows the hitcount values (left/right), co-hitcount values (center) and the number of pages containing only one of two topics (left-centered/right-centered).

our analysis showed that it suffices for many cases and gives a useful base for the more complex models described in the following sections.

2.2 Co-visibility

To be able to describe dependencies between different topics we introduce the measure of co-visibility of two topics[4] based on co-occurrence: Two topics top_1 and top_2 co-occurring in a large number of documents should have *something* in common.[5] We measure the co-occurence with a co-hitcount value which we define as the hitcount of a search engine when searching for "top_1 AND top_2" (Fig. 5).

Again, an example from [4] illustrates how the co-hitcount of two topics can be used in a marketing scenario: in August 2005, all German carriers in the mobile phone market started offering flatrate contracts[6], called 'handyflatrate' (the German term for a mobile phone is 'handy'). Figure 6 shows the visibility of 'handyflatrate' from Aug. 3 to Aug. 24: it doubled in the beginning and returned to a hitcount of around 150.[7]

Figure 7 shows the co-hitcounts of the German main carriers T-Mobile, Vodafone, O2, E-Plus and Debitel with 'handyflatrate': all carriers gained visibility and it is obvious that the three biggest carriers generally had the highest values. However, the curve of E-Plus grows steadily and almost reaches that of the very big carrier Vodafone, while all other carriers settle on their level or even decrease. The same is revealed by Fig. 8 comparing the relative co-hitcounts ('attention shares') for carrier plus 'handy' and carrier plus 'handyflatrate' on Aug. 25. Note

[4] We restrict ourselves to two topics, generalization for three or more topics is possible.
[5] Whatever this "something" is. It is often *not* semantic closeness for authors not necessarily use synonyms within one text. So the interpretation of co-visibility has to be left to the user.
[6] 'Pay a constant amount of money per month and phone as long as you want'.
[7] All these monitorings were restricted on the domain 'de' to focus on the German market.

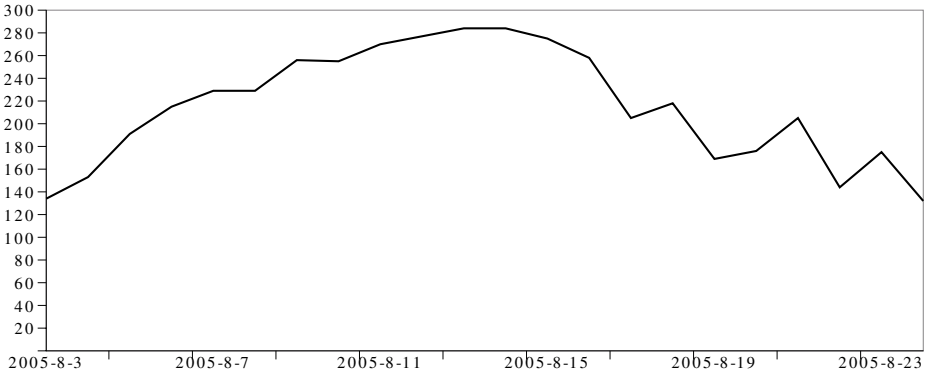


Fig. 6. Estimated hitcount (Google) for 'handyflatrate' on the domain 'de' in time

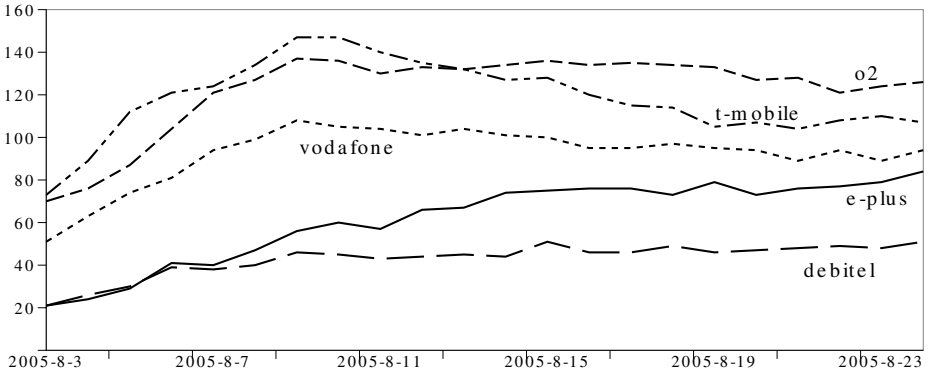


Fig. 7. Estimated co-hitcount (Google) for different carriers and 'handyflatrate' on the domain 'de' in time

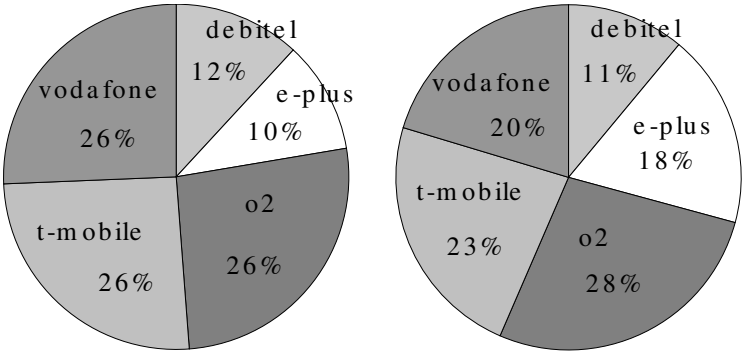


Fig. 8. Percentual co-visibilities of carriers and 'handy' (left), carriers and 'handyflatrate' (right) on the domain 'de' on Aug. 25

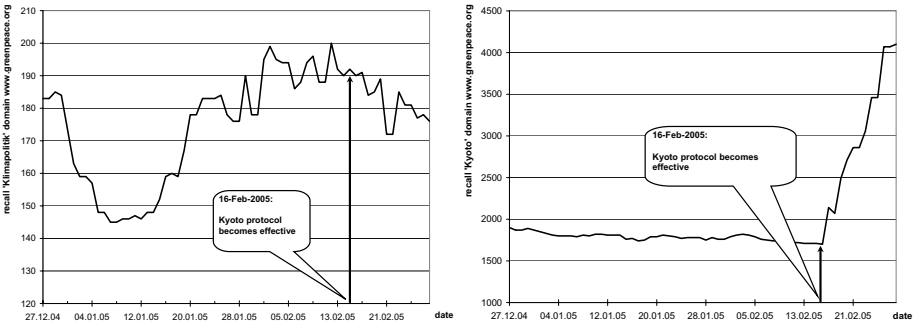


Fig. 9. Estimated hitcount values (Google) for 'Klimapolitik' (climate policy) [left] and 'Kyoto' [right] on `www.greenpeace.org`

that we could not use pure carrier hitcounts because of the special string 'O2' (we do not want to count pages related to oxygen). This diagram shows that E-Plus could increase their co-hitcount in the field of handyflatrates, compared to the overall hitcount of the company itself. This example illustrates how co-hitcounts can be used to link products with companies to analyze a market with different competitors under the aspect of public attention.

Anyhow, for many applications not the total number of pages is of interest, but the ratio between the number of pages containing both topics and the number of pages containing at least one of them. So we define

$$\mathrm{covis}_i(top_1, top_2) \quad = \quad \frac{\mathrm{cohitcount}(\text{“top}_1\text{”}, \text{“top}_2\text{”})}{\mathrm{hitcount}(\text{“top}_i\text{”})}, \qquad i \in \{1, 2\}$$

which allows us to determine the degree of connection between several terms (currently or monitored in time).

3 Semantic Relations Between Topics

3.1 The Insufficiency of the Simple Visibility Measure

We tracked our example of climate policy in the domain `www.greenpeace.org` some further weeks and expected a rise in visibility on Feb. 16, 2005. At that date, 90 days after the ratification by Russia, the Kyoto protocol became effective. We expected important events like this to stimulate discussions on the topic climate policy and to be measurable in a domain dealing with environmental protection. Our results, pictured in Fig. 9 [left], did not support this hypothesis.

Contrariwise, the right side of Fig. 9 evidences an immense visibility gain for the topic 'Kyoto' in the same domain. This is easy to explain: an author writing an article for `www.greenpeace.org` reporting on the latest news on the Kyoto protocol will not necessarily use the phrase 'Klimapolitik', but definitely the word 'Kyoto'. On the other hand, doing without 'Klimapolitik' and monitoring only

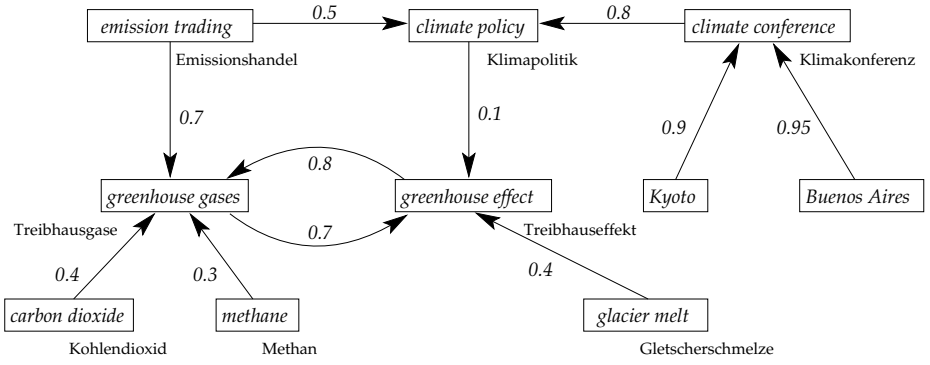


Fig. 10. Semantic network for topics of climate policy

'Kyoto' will not work likewise, for we cannot know a priori what *will* happen, so monitoring of at least the two topics 'Klimapolitik' and 'Kyoto' seems advisable. In general, this demonstrates the necessity to monitor more than one topic, more precisely several topics that are semantically related.

3.2 Semantic Network of Topics

We represent the following kind of relation between topics: two topics are semantically related, if the visibility of one topic automatically raises the visibility of the other. In other words: If a discussion on top_1 to a certain degree automatically concerns top_2, we designate top_1 as semantically related to top_2. Additionally, a weight $\mathrm{W}(top_1, top_2) \in [0, 1]$ qualifies the closeness of each relation with high values denoting a close relation. Take the topics HIV and aids as an example: A discussion on aids almost always also concerns HIV, for aids is always caused by the HI-virus. Actually, the two terms are quite often used synonymically. Further on, in the context of an environmental website, the topic Kyoto will rather reference the topic climate conference than the city of Kyoto, so a high semantical relation from Kyoto to climate conference exists. Note that our concept of semantical relationship is not symmetrical, e. g. a discussion on climate conference does not automatically as well concern Kyoto. Modeling the relations between several topics, we obtain a directed and weighted graph of topics, like illustrated in Fig. 10 for our example of climate policy. This graph corresponds to the well-known concept of semantic networks[8]. Keep in mind that the modeling of semantic topic networks certainly heavily depends on the context and the view of the modeler and cannot be specified objectively. In the case of the 0.9 between 'Kyoto' and 'climate conference' in Fig. 10, for example, this weight seems much too high for Kyoto might also refer to normal pages of the city Kyoto. But in the context of the domain www.greenpeace.org, Kyoto will almost always refer to a climate conference.

[8] See [5] for a comprehensive reading.

Although we regard visibilities as a general concept, the interpretation of visibilities as hitcount values like introduced in section 2 yet makes things clearer: An edge with weight $\mathrm{W}(\textit{Klimakonferenz}, \textit{Klimapolitik}) = 0.8$ claims that 80% of the web pages containing the string 'Klimakonferenz' as well concern the topic climate policy. Note that this is not a statement on co-visibility, i.e. those 80% may but need not necessarily contain the string 'Klimapolitik', but the results of a co-visibility request might help to build up the semantic network.

4 Spreading Activation with Co-visibilities

The algorithms we present in this section are based on the Spreading Activation algorithm (SA). SA was first introduced by psychologists as early as in the 1960's (see e. g. [6,7]) to explain human associative memory. Recently, SA was adopted for propagation of trust between actors in trust networks [8]. Furthermore, SA was utilized to improve methods in information retrieval (see e. g. [9,10,11]). The basic idea of SA is that of energy flowing through a network along weighted edges. Lausen and Ziegler specify the algorithm recursively (Alg. 1).

Algorithm 1 Spreading activation algorithm by Lausen and Ziegler [8].

procedure energize($e \in R_0^+, s \in V$) {

$\quad \mathrm{energy}(s) \longleftarrow \mathrm{energy}(s) + e$;

$\quad e' \longleftarrow \dfrac{e}{\sum_E \mathrm{W}(s,n)}$;

$\quad$ **if** $e > T$ **then** $\forall (s,n) \in E : \mathrm{energize}(e'\,\mathrm{W}(s,n), n)$;

}

V denotes the set of all nodes, E the set of all edges, s the node that is energized, e the amount of energy pushed into node s, $\mathrm{energy}(s)$ a data structure holding the current energy for each node (0 in the beginning), $\mathrm{W}(s,n)$ the weight of the edge from node s to node n. The energy a node s receives during one call of energize is disseminated proportionally on all outgoing edges of the node, depending on the accordant weight of the edge. This assures that not more energy than the injected energy e will leave the node. All nodes with incoming edges

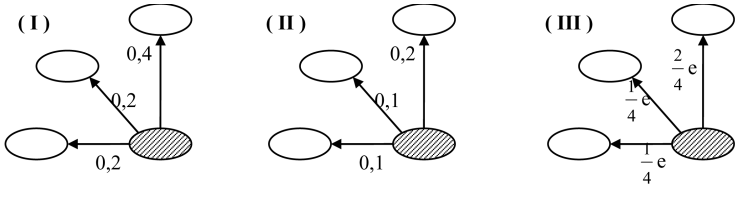


Fig. 11. Standard SA with energy spreading from the gray node

Algorithm 2 Spreading Activation algorithm for visibility adjustment

procedure visibilize($v \in R_0^+, t \in V$) {
  $\text{vis}(t) \leftarrow \text{vis}(t) + v$; ;
  if $v > T$ **then** $\forall (t,n) \in E : \text{visibilize}(v\,\text{W}(s,n), n)$;
}

from s are energized by a recursive call. Thus, energy packages with decreasing size flow through the network until their size falls under a certain threshold T and the algorithm terminates.

For the problem of visibility adjustment, a modification of this algorithm becomes necessary: Through the normalization of the outgoing energy, the graphs (I) and (II) in Fig. 11 become equivalent. This is contradictory to our intuition that a high semantic closeness between two topics should make more energy flow. Secondly, the assumption of SA that energy may not come from nothing, i. e. not more energy may leave a node than has been injected, is obsolete for visibilities. In fact, the notion of web pages concerning other pages implies some kind of 'hidden' visibility we strive to extract with our algorithm, so that a visibility gain is intended. We therefore simplify algorithm 1 and obtain algorithm 2, called visibilize for topic t and visibility v.

Algorithm 3 Spreading Activation algorithm with co-visibilities (1st version)

procedure visibilize($v \in R_0^+, t \in V$) {
  $\text{vis}(t) \leftarrow \text{vis}(t) + v$; ;
  if $v > T$ **then** $\forall (t,n) \in E : \text{visibilize}(v\,\text{W}(s,n)(1 - \text{covis}_1(t,n)), n)$;
}

Algorithm 4 Spreading Activation algorithm with co-visibilities (2nd version).

procedure visibilize($v \in R_0^+, t \in V, top_S \in V$) {
  $\text{vis}(t) \leftarrow \text{vis}(t) + v$; ;
  if $v > T$ **then** $\forall (t,n) \in E : \text{visibilize}(v\,\text{W}(s,n)(1 - \text{covis}_1(top_S,n)), n, top_S)$;
}

Using this algorithm, an adjustment of visibility is achieved as follows: model the semantic network of topics. Acquire the initial visibilities like described in section 2. For each topic t in the network call visibilize(t, v_{init}) with the initial visibility v_{init} of topic t, see Fig. 12 for an example with three topics and initial visibilities 100, 50, 10.

4.1 Spreading Activation with Co-visibilities

We do not settle for Algorithm 2, but improve it by adding knowledge from the co-visibilities. Imagine top_1 and top_2 from Fig. 12, with their initial visibilities of 100 and 50, having a $\text{covis}_1(top_1, top_2)$ of 0.4 and a $\text{covis}_1(top_2, top_1)$ of 0.8. In other words: 60 pages contain only the string of top_1, 10 pages only top_2, 40 pages contain both strings. Spreading the visibility of 50 from top_2 to top_1 and a visibility of 100 from top_1 to top_2 is not appropriate in this case, for some visibility would be counted double. We avoid this by introducing co-visibilities into our algorithm, refer to algorithm 3. Effectively, we adjust the weights of the net. Note that this adjustment is different for each date of monitoring, because the co-visibilities differ from day to day, while the original weights in the semantic network express the closeness of the relation and remain constant over time. Fig. 13 illustrates the first visibilization step for the new algorithm.

One aspect that is not covered by algorithm 3 is how to take cyclical and transitive relations into account for co-visibilities: In algorithm 3 we use for each package of energy propagated along an edge from top_1 to top_2 the co-visibility between top_1 and top_2. A more sophisticated strategy would take the co-visibility between the source topic top_S, i. e. the topic where the visibility has been injected, and top_2 (algorithm 4). The initial call of visibilize is executed with $t = top_S$. The recursive calls hand on the source-parameter and always use the co-visibility between source and the current target topic.

Going back to our example of climate policy, we run algorithm 4 on the initial visibility data of Fig. 9 with the semantic network of Fig. 10. We obtain adjusted visibilities for 'Klimapolitk', the topic we are interested in. Fig. 14 displays the

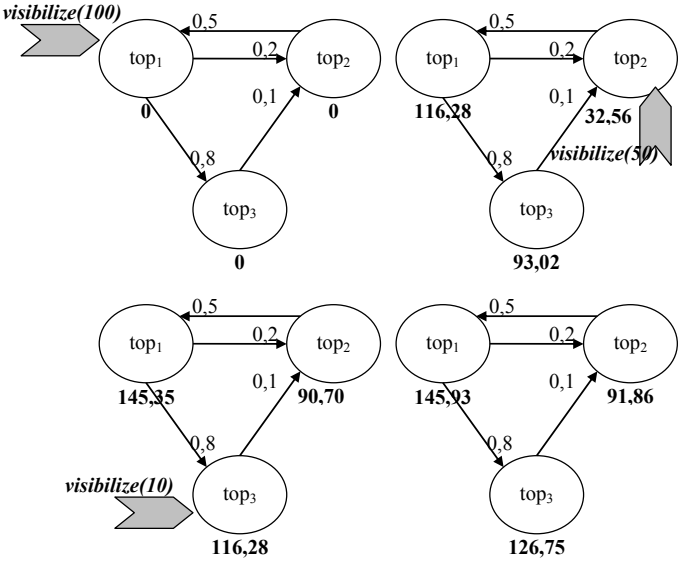


Fig. 12. Injection of visibility into a semantic network with three calls of algorithm 2

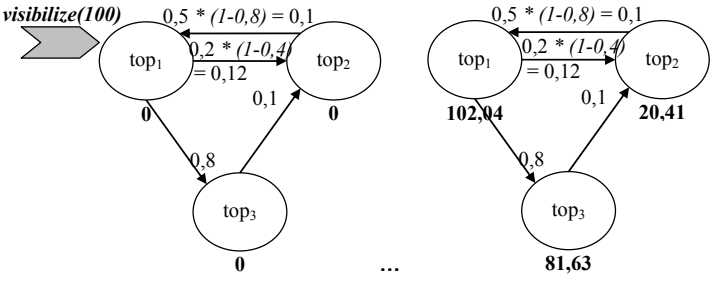


Fig. 13. Injection of visibility into a semantic network: first step with call of algorithm 3

initial visibilities of 'Klimapolitik' (lower curve), the initial visibilities of 'Kyoto' (center curve) and the adjusted visibilities of 'Klimapolitik' (upper curve). The developing of 'Klimapolitik' adapts itself to the developing of 'Kyoto'. This is no surprise, because we chose quite high weights in our semantic net leading to large packages of visibility flowing through the net.

5 Reliability of Search Engine Hitcounts

As mentioned in section 2, we obtained our hitcounts from the Google Web API, although the algorithms shown in this paper are not dependent on Google hitcounts, but also work with other sources of visibility. When we started our research on visibility in November 2004, the Google Web APIs seemed to be appropriate because of easy usability. However, Google state in their terms and

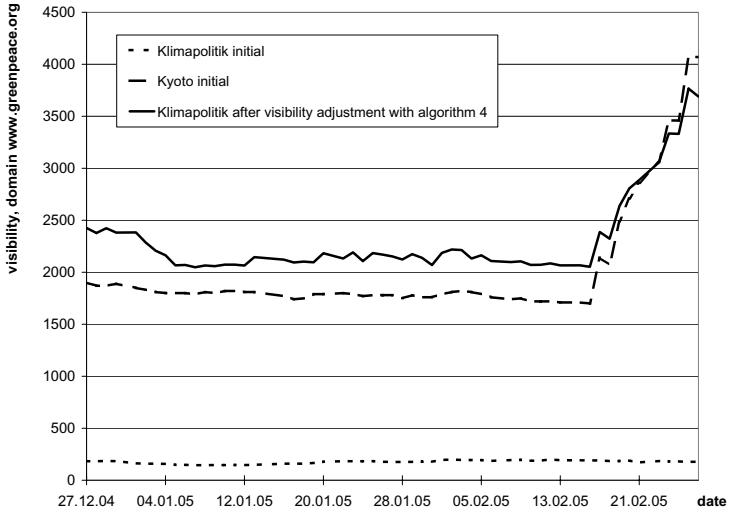


Fig. 14. Initial visibilities from Fig. 9 and adjusted visibilities for 'Klimapolitik' using the semantic network of Fig. 10 with Algorithm 4

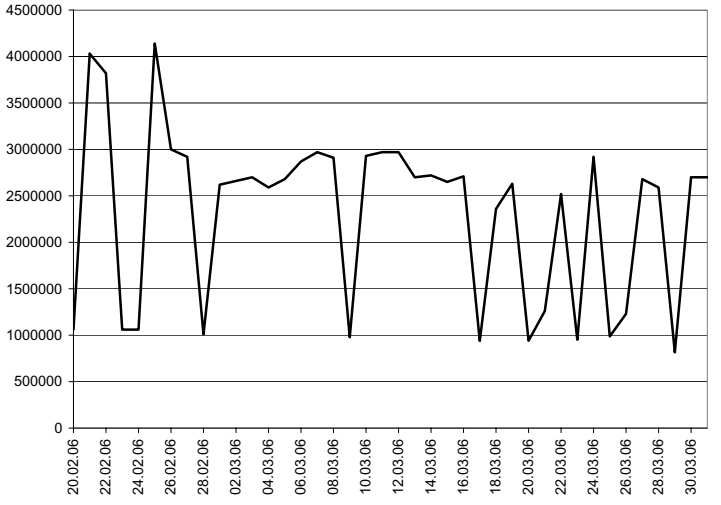


Fig. 15. Estimated hitcounts (Google API) of Dänemark (denmark)

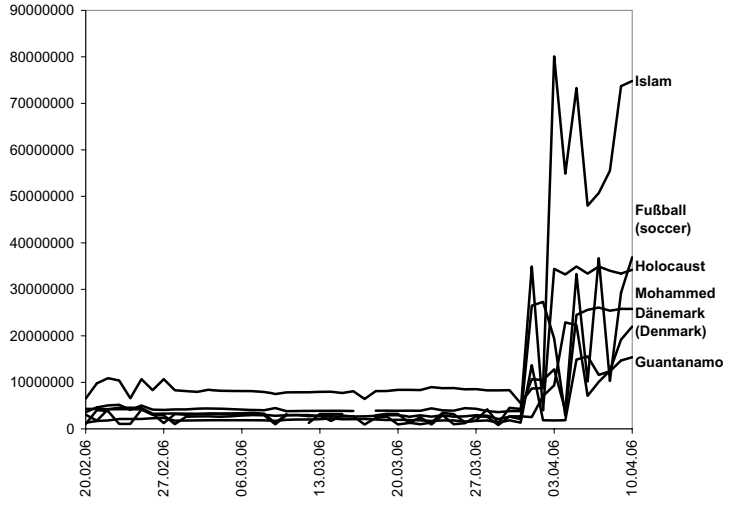


Fig. 16. Estimated hitcounts (Google API) of several topics increase tenfold on 1 April 2006

conditions for Google Web API service: 'The Google Web APIs service is currently in beta form and has not been fully tested or debugged' [9]. In general, using a search engine whose mechanisms you do not know in detail always implies relying on a black box. Other papers have reported on drawbacks of the

[9] `http://www.google.com/apis/api_terms.html`

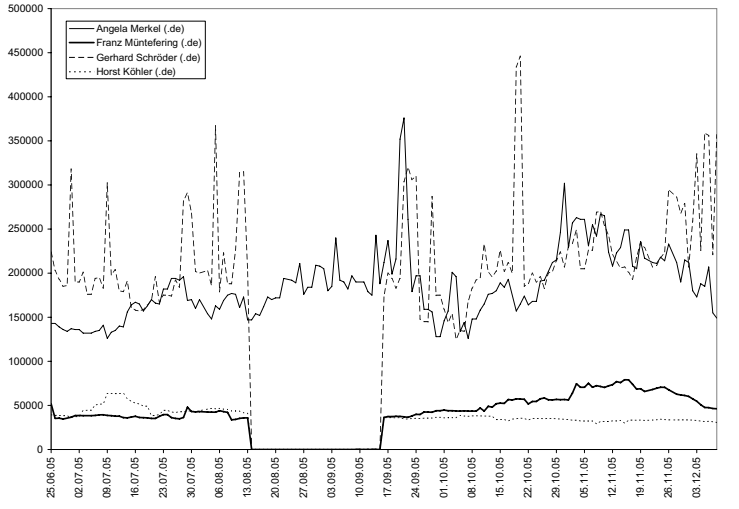


Fig. 17. Estimated hitcounts (Google API) of politicians with German umlauts

search engine's API, e.g. Mayr and Tosque who state that the hitcounts returned by the standard Google web interface and those from the Google API differ (see [12]). In his blog, Prof. Jean Véronis elaborates on some other inconsistencies of Google's hitcounts (see [13] and follow-up messages): for example, in January 2005 Google counted 8.000.000.000 pages containing 'the' on the whole internet, and 88.100.000 pages with 'the' on the English speaking web. That would imply that 99% of the pages containing 'the' were in other langugages than English.

In the course of our research, we experienced some further problems which come along with the Google Web API and which have a high impact on the API's reliability.

Consider Fig. 15 showing the hitcounts for 'Dänemark' (Denmark) in the course of one month: the values keep jumping from around 3.000.000 to 1.000.000 and back all the time. It is unrealistic to believe that the web has grown and shrunk so fast, so we can assume that some values are erroneous. Problems like this occured in some of our curves, especially in those with frequent topics like 'Denmark' or 'terrorism'. Although we knew a priori that we will not retrieve exact hitcounts, variations of this kind can hardly be seen as simply caused by estimation.

Figure 16 shows another problem which occured around Apr. 1, 2006: suddenly the hitcounts increased by factor ten (Fig. 16 only shows a small selection of the topics currently monitored). Finally, we had problems concerning German umlauts (ä, ö, ü) in August 2005: all topics with one of those letters dropped to almost zero for one month. This was particularly annoying because we monitored the hitcounts of German politicians like 'Schröder' (see Fig. 17) to evaluate the German elections in September 2005.

To cope with these problems hitcount data has to be handled carefully:

- A number of (independent) topics has to be monitored simultaneously to be able to detect hitcount changes caused by search engine internals (which should affect all topics).
- For many curves show dropouts (e. g. Fig. 15), single values are not reliable, so the average curve progression has to be used.
- Monitoring the hitcount of the same topics with more than one search engine can help to minimize the risk of data loss caused by errors like that of Fig. 17.
- No search engine is able to crawl the whole web and hitcount values are estimated, so the absolute numbers are rather erroneous. Therefore, the hitcount must not be interpreted absolutely. Instead, the relative change in time or the relation to the hitcounts of other topics should be used.
- The monitoring has to be supervised to detect failures like the umlaut problem, so the data should not be given to an uninformed end user.

6 Related Work and Conclusions

The first emphasis of our paper were the examples showing that real world events have an impact on the visibility of topics on the web. One problem yet remaining unresolved in this context is that, in contrary to our example of Kyoto, we often cannot predict which events could occur and which topics would be interesting to monitor. The terrorist attack of September 11, for example, surely had an effect on the visibility of 'terrorism' or 'World Trade Center', but nobody could know in advance that a monitoring of these topics would be interesting. The moment the event happens, the historical data is missing. A possible solution could be the usage of historical data from communication processes with timestamp, e.g. from a discussion group.

Analyzing the dynamically changing web has been done quite often: [14], for example, investigate the correlation between age of web pages and their quality to improve PageRank, while [15] monitor changes on the web to estimate the rate for reasonable search engine re-indexing. To our knowledge, no approach to correlate visibility and real world events exists.

In a second step, we modeled semantic relations between topics in semantic networks to add prior knowledge to our visibility analysis. Although these semantic networks look similar to Bayes Networks [16], Bayes networks do not permit the cyclic relations we need for modelling the semantic closeness of topics.

The approach of semantic networks was chosen to keep the algorithms simple. Nevertheless, an approach like thesauri with more than one type of relation would offer a much more intuitive modeling and therefore save time. In [17], a semi-automatical derivation of a semantic network from a user-modeled thesaurus is proposed. This would combine the intuitive modeling of thesauri with the convenience of a relatively simple algorithm for semantic networks.

The third input to our algorithm, besides visibilities and a semantic network, were co-visibilities. A possible application of co-visibilities we did not address in our paper is the automatic extraction of facts. This has recently been done

by Etzioni et al. who used hitcount values from a search engine for their system called KnowItAll to automatically extract facts from the WWW [18]. Search engine queries were also used by [19] for an automatical detection of synonyms and by [20] for the validation of question-answering systems, which both are further areas of application for co-visibilities. Co-occurences of terms are visualized in [21] for identifying significant topics in corpora of daily news.

We plan to endorse our findings on the relation between real world events and visibility with larger case studies. Investigations of at least more than one year should prove the applicability of visibility analysis in the long-term.

Finally, we intend to integrate the concept of visibility of topics into communication oriented modeling (COM) [22]. COM investigates large-scale communication processes with message/reference-networks like internet discussion groups. A definition of the concept of topic visibility for this kind of communication processes could be made. With the COM testing environment (COMTE)[10], further analysis could reveal correlations between author visibility, message visibility, topic visibility and the structure of the reference network.

References

1. Gans, H.J.: Deciding What's News: A Study of CBS Evening News, NBC Nightly News, 'Newsweek' and 'Time'. 25th anniversary edn. Northwestern University Press, Evanston, IL (2005)
2. Yungwook, K.: Measuring the economic value of public relations. Journal of Public Relations Research **13** (2001) 3–26
3. Falkinger, J.: Attention economies. CESIFO WORKING PAPER NO. 1079, ifo Institut für Wirtschaftsforschung, München (2003)
4. Kiefer, P., Stein, K.: Visibility analysis on the web as an indicator for public relations and marketing evaluation. In: Proc. of Intelligent Agents, Web Technology and Internet Commerce (IAWTIC 2005), IEEE Computer Society Publishing (2005)
5. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA (2000)
6. Quillian, R.: Semantic memory. In Minsky, M., ed.: Semantic Information Processing. MIT Press, Boston, CA, USA (1968) 227–270
7. Collins, A.M., Loftus, E.F.: A spreading-activation theory of semantic processing. Psychological Review **82** (1975) 407–428
8. Lausen, G., Ziegler, C.N.: Spreading activation models for trust propagation. In: IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE '04. (2004) 83–97
9. Preece, S.: A spreading activation network model for information retrieval. PhD thesis, CS Dept., Univ. of Illinios, Urbana, IL. (1981)
10. Crestani, F.: Applications of spreading activation techniques in information retrieval. Artificial Intelligence Review **11** (1997) 453–482
11. Ceglowski, M., Coburn, A., Cuadrado, J.: Semantic Search of Unstructured Data using Contextual Network Graphs. National Institute for Technology and Liberal Education (2003)

[10] `http://www.kinf.wiai.uni-bamberg.de/COM/`

12. Mayr, P., Tosques, F.: Google Web APIs - an instrument for webometric analyses? Poster presented at ISSI 2005 (2005) http://arxiv.org/ftp/cs/papers/0601/0601103.pdf.
13. Véronis, J.: Web: Google's counts faked? Blog, see also follow-up messages (2005) http://aixtal.blogspot.com/2005/01/web-googles-counts-faked.html.
14. Baeza-Yates, R., Saint-Jean, F., Castillo, C.: Web structure, age and page quality. In: Proceedings of the 2nd International Workshop on Web Dynamics (WebDyn 2002). (2002) http://www.dcs.bbk.ac.uk/webDyn2/onlineProceedings.html.
15. Brewington, B.E., Cybenko, G.: How dynamic is the Web? Computer Networks (Amsterdam, Netherlands: 1999) **33** (2000) 257–276
16. Russel, S., Norvig, P.: Chapter 14, Probabilistic Reasoning. In: Artificial Intelligence: A Modern Approach. Prentice Hall (2003) 492–536
17. Kiefer, P.: Computational analysis of the visibility of themes in internet-based communication processes, in German: Softwaregestützte Analyse der Sichtbarkeit von Themen in internetbasierten Kommunikationsprozessen. Diploma thesis, Chair for Computing in the Cultural Sciences, Bamberg University, Bamberg (2005)
18. Etzioni, O., Cafarella, D., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence (2005) 91–134
19. Turney, P.: Mining the web for synonyms: Pmi-ir versus lsa on toefl. In: Proceedings of ECML2001, Freiburg, Germany (2001) 491–502
20. Magnini, B., Negri, M., Tanev, H.: Is it the right answer? Exploiting web redundancy for answer validation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. (2002) 425–432
21. Richter, M.: Analysis and visualization for daily newspaper corpora. In: Proc. of RANLP. (2005) 424–428
22. Malsch, T., Schlieder, C., Kiefer, P., Lübcke, M., Perschke, R., Schmitt, M., Stein, K.: Communication between process and structure: Modelling and simulating message-reference-networks with COM/TE. Journal of Artificial Societies and Social Simulation (2006) accepted.

Link-Local Features for Hypertext Classification

Hervé Utard and Johannes Fürnkranz

TU Darmstadt, Knowledge Engineering Group
Hochschulstraße 10, D-64289 Darmstadt, Germany
herve.utard@ingenieurs-supelec.org,
juffi@ke.informatik.tu-darmstadt.de

Abstract. Previous work in hypertext classification has resulted in two principal approaches for incorporating information about the graph properties of the Web into the training of a classifier. The first approach uses the complete text of the neighboring pages, whereas the second approach uses only their class labels. In this paper, we argue that both approaches are unsatisfactory: the first one brings in too much irrelevant information, while the second approach is too coarse by abstracting the entire page into a single class label. We argue that one needs to focus on relevant parts of predecessor pages, namely on the region in the neighborhood of the origin of an incoming link. To this end, we will investigate different ways for extracting such features, and compare several different techniques for using them in a text classifier.

1 Introduction

Whereas the exploitation of the graph properties of the Web is already standard in search engine technology (and has resulted in a major break-through with the advent of Google's page-rank algorithm), its use for improving hypertext classification is still a hot research topic [5]. Conventional approaches to link-based text classification use either the entire text of a predecessor page or abstract this information into a class label for the entire page. Our working hypothesis is that the first approach is unsatisfactory, because not the entire page of a preceding link is relevant, and the second approach is too coarse because predecessor pages may be about entirely different topics, and may not fit into the predefined classification scheme. Nevertheless, it is clear that some *part* of a preceding page must contain information about the target page because there is a hyperlink connecting these two pages, and this link is (typically) annotated with its anchor text or text in the proximity of the hyperlink.

In this paper, we try to exploit this information by capturing different kinds of proximity (both at the structural and the textual level). We propose different types of features that can be extracted from HTML-documents and evaluate their utility for hypertext classification. Our results will demonstrate an improvement for several types of features (e.g., words in the neighborhood of a hyper-link), which could not be observed in previous work where the entire text of predecessor documents was used. Other features (e.g., headings) seem to be primarily useful in combination with others.

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 51–64, 2006.

2 Hypertext Classification

It has been recognized early on that hyperlinks may provide important information about the content of a Web page. Anchor texts (texts on hyperlinks in an HTML document) of predecessor pages were already indexed by the WWW Worm, one of the first search engines and web crawlers [10]. Later, the importance of predecessor pages for ranking search results was established with the striking success of Google's page rank [1] and related techniques such as the HITS algorithm [8].

Not surprisingly, recent research has also looked at the potential of hyperlinks as additional information source for hypertext categorization tasks. Many authors addressed this problem in one way or another by merging (parts of) the text of the predecessor pages with the text of the page to classify, or by keeping a separate feature set for the predecessor pages. For example, Chakrabarti et al. [2] evaluate two variants: (1) appending the text of the neighboring (predecessor and successor) pages to the text of the target page, and (2) using two different sets of features, one for the target page and one resulting from a meta-document that is a concatenation of the neighboring pages. The results were negative: in two domains both approaches performed worse than the conventional technique that uses only features of the target document.

Chakrabarti et al. [2] concluded that the text from the neighbors is too unreliable to help classification. They proposed a different technique that included predictions for the class labels of the neighboring pages into the model. As the labels of the neighbors are not known, the implementation of this approach requires an iterative, EM-like technique for assigning the labels, because changing the class of a page may potentially change the class assignments for all its neighboring pages as well. The authors implemented a relaxation labeling technique, and showed that it improves performance over the standard text-based approach that ignores the hyperlink structure.

Lu and Getoor [9] generalize this work by enhancing the set of features that are considered for each neighboring page. In particular, they distinguish between in-neighbors, out-neighbors and co-linked neighbors. An in-neighbor, a *predecessor*, is a web page that contains a link pointing to the target page; an out-neighbor, a *successor*, is a web page that is linked by the target page; and the co-linked neighbors are web pages which have a common in-neighbor. Their conjecture is that cumulative statistics on the neighbors' class distribution are as informative as the identity of the neighbors, which requires much more storing space. An interesting characteristic of their model is that instead of going with the majority prediction, they learn how the category distribution of the neighbors affects the prediction. Like Chakrabarti et al. [2], they employ an iterative relaxation labeling technique, and compare two classifier types: a flat model where the local features and the non-local ones are concatenated into a common vector. The local and non-local features are thus not distinguished. The second model is obtained by combining the predictions of both a classifier based on the local features and a classifier based on the non-local features. The flat model is outperformed by the second one, which confirms the results of Chakrabarti.

3 Link-Local Features

The approaches described in the previous section basically employ two techniques for using the information on predecessor pages:

- use all words in predecessor documents as features
- abstract the predecessor documents into a single feature (the class label)

The main hypothesis of our work is that the first approach contains too much irrelevant information, whereas the second approach throws away too much relevant information: Using the entire text of the predecessor pages loses the focus on the relevant parts of the predecessor documents. For example, a page may be predecessor to several pages, each of which may belong to a different class. Thus, if the entire text is used, each example would be represented in the same way, and discrimination would be impossible. Focusing on the region around the respective hyperlinks, would result in different feature sets. On the other hand, abstracting the entire information on the page into a single class label is, again, problematic, for several reasons. First, a single page will always have the same class label, which may, again, lead to problems such as those discussed above (different pages that share the same set of predecessors are indiscriminable). Second, an important assumption that the approaches of Chakrabarti and Getoor make is that all (or a significant number) of the neighboring pages of a page can also be classified into the same set of classes. This, however, need not be the case. If you, for example, have the task of classifying web-pages about music between *official artist websites* and *fans websites*, the majority of the predecessors of an official website will belong to the *fans websites* set.

On the other hand, even if the entire page cannot be classified into the available set of classes, it can be expected that the parts of the predecessor pages around the hyperlink to the page in question contain relevant information. For example, Figure 1 shows three different feature types that could be important for classifying a target page: the anchor text of a page (left), the text in a paragraph around the hyperlink (center), and the text in a heading preceding the hyperlink (right). In all three cases, the text allows to classify the target page as a *professor*-page, whereas this cannot be inferred from the text on the target page itself. Thus, features that occur in a textual or structural proximity to the relevant out-link can be of importance. We will call such features *link-local*.

In previous work [6], we tried to utilize link-local features with so-called *hyperlink ensembles*. The idea is quite simple: instead of training a classifier that classifies *pages* based on the words that appear in their text, a classifier is trained that classifies *hyperlinks* according to the class of the pages they point to, based on the words that occur in their neighborhood of the link in the originating page. In the simplest case, we use the anchor text of the link. Consequently, each page will be assigned multiple predictions for its class membership, one for each incoming hyperlink. These individual predictions are then combined to a final prediction by some voting procedure. Thus, the technique is a member of the family of ensemble learning methods [4]. In a preliminary empirical evaluation

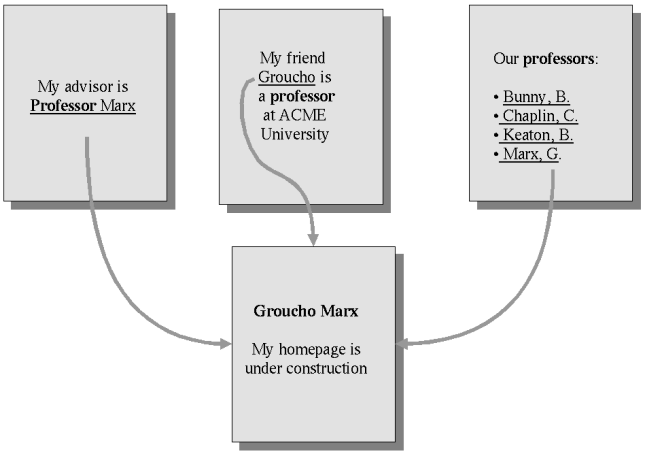


Fig. 1. Three different feature types on predecessor pages that might be helpful for classifying a page: anchor text, text in a paragraph, text in a preceding heading

in the Web→KB domain (where the task is to recognize typical entities in Computer Science departments, such as faculty, student, course, and project pages), hyperlink ensembles outperformed a conventional full-text classifier in a study that employed a variety of voting schemes for combining the individual classifiers and a variety of feature extraction techniques for representing the information around an incoming hyperlink (e.g., the anchor text on a hyperlink, the text in the sentence that contains the hyperlink, or the text of an entire paragraph). The overall classifier improved the full-text classifier from about 70% accuracy to about 85% accuracy in this domain.

The main point of this paper is to investigate this approach in more detail. Most importantly, we will demonstrate that these feature extraction techniques will result in even better performance on a conceptually simpler approach that combines all features into a single predecessor document.

4 Link-Local Features Extracted

As with classical text classification, we extract word-based features from the documents. We test different heuristic patterns to target the words which give the most relevant information about each link, namely the *anchor text*, the *paragraph around the link*, the *words neighboring the anchor*, the *headings structurally preceding the link*, *the heading of the list*, if the link is part of an HTML list and the *text of the document to classify*.

In particular, we extract the following features:

Link Description: The first spot is the *link description*, also named *anchor text*. It consists of the text that occurs between the HTML Tags `<A HREF=...>` and `</A>` of the link pointing to the page to classify.

Link Paragraph: The paragraph around the anchor text may also contain interesting words that describe the target page. We extract it in the features group *Link Paragraph*. We use the HTML tags `<P>` and `</P>` to find the borders of the paragraph.

Link Neighborhood: One difficulty with *Link Paragraph* is that the size of the paragraphs varies. The purity or the dilution of the clue features in the crowd of the words is not constant. We circumvent this problem with the features group *Link Neighborhood* where a fixed number of words neighboring the link are mined. The link description is excluded from *Link Neighborhood*. This feature location is an important source of information for the links with an irrelevant anchor text like "click here" or "next page".

Link Headings: In the *Link Headings* features, we extract the words occurring in the headings *structurally* preceding the link in the predecessor. We consider headings of type `H1`, `H2`, and `H3`.

Link List Headings: Sometimes, the link is part of an HTML list (tag `<UL>`). In this case, we store the preceding heading of each depth in the features group *Link List Headings*.

Own Text: The last but simplest feature set is the text of the target page itself. We extract it mainly in order to compare our model with traditional text-based classifiers.

These features are essentially the same that were suggested in [6]. However, in this work we extracted them in a more principled way using XPath structural patterns on the Document Object Model (DOM) representation of the documents. XPath is a language for navigating through elements and attributes on an XML document. It uses path expressions to select nodes or node-sets in an XML document. These path expressions are similar to those used for locating files in a file system. Table 1 lists the XPath expressions we use to extract the features from the predecessors of the target document.

The result of the *Link Description* request is the concatenation of the segments of the HTML file that occur between the HTML tags `<A HREF='Target_URL'>` and `</A>`. The other requests are simple extensions of the *Link Description* request. Once the anchor tag of the links is localized, *Link Paragraph* looks for its last ancestor of type Paragraph. *Link Headings* looks for the last occurrence of each heading level before the link, and *Link List Headings* looks for the last occurrence of each heading level before the beginning of the list. The implementation can certainly be improved, e.g., by trying to recognize headings that are not formatted with `<H?>` tags. For example, text immediately preceding an `<UL>` list might be interpreted as a heading for this list. However, we preferred to rely on the information that is directly provided by the HTML structure.

As HTML is not XML compliant, we first translate the web pages from HTML format into XHTML format with the help of the Tidy program. We encountered a problem by this step because some HTML pages contain many syntax errors. Tidy cannot understand them all and can thus not output the XHTML translation of all the documents. We circumvent this difficulty by extracting the basic features (text of the target page) on the HTML page before the Tidy treatment

Table 1. XPath expressions for extracting the features

Link Description	`//a[\@href='Target_URL']`
Link Paragraph	`//a[\@href='Target_URL']/ancestor::p[last()]`

Link Headings	`//a[\@href='Target_URL']/preceding::h1[last()]`
	`| //a[\@href='Target_URL']/preceding::h2[last()]`
	`| //a[\@href='Target_URL']/preceding::h3[last()]`

Link List Headings	`//a[\@href='Target_URL']/ancestor`
	`::ul/preceding::h1[last()]`
	`| //a[\@href='Target_URL']/ancestor`
	`::ul/preceding::h2[last()]`
	`| //a[\@href='Target_URL']/ancestor`
	`::ul/preceding::h3[last()]`

and the complex features (link description, headings, ...) after the construction of the DOM tree by Tidy. If Tidy fails to convert HTML to proper XHTML, these features will not be extracted from this predecessor. As a page typically has several predecessors from which features are extracted, we can afford to do this.

5 Experimental Setup

5.1 Datasets

Allesklar (`http://www.allesklar.de`) is a German generic web directory referencing about 3 million of German web sites. Its tree organization begins with 16 main category roots, each one containing between 30,000 and 1,000,000 sites. The nodes of the tree are as specific categories as the node is deep. We chose 5 main categories, shown in Table 2.

We crawled each selected category with a breadth-first traversal in order to collect pages covering the whole category. We looked for hyperlink predecessors for each of these pages using the Altavista `link` request (for example, the request `link:europa.eu.int` retrieves all the web sites containing a link to the Web portal of the European Commission).

We looked for up to 25 predecessors per example. But we couldn't always find as many predecessors and some predecessors referenced by Altavista were not always reachable. However, only a small part of the examples have no predecessor and a large part of them has more than 10 predecessors. There is no important difference between the categories from this point of view. Only the distribution of *Immobilien & Wohnen* is slightly biased towards fewer predecessors.

The *Web→KB dataset* [3] is a collection of web pages coming from the science departments of four main universities: *Cornell*, *Texas*, *Washington* and *Wisconsin*. One fifth group of pages named *misc* has been collected from various other universities. These pages are classified under seven categories: *course*, *department*, *faculty*, *project*, *staff*, *student* and *other*. The WebKB dataset is not equally distributed (Table 3): more than 45% of the examples are concentrated

Table 2. Class distribution for the Allesklar dataset

Category		3898 Examples
Arbeit & Beruf	(Employment & Jobs)	601 (15.42%)
Bildung & Wissenschaft	(Education & Science)	849 (21.78%)
Freizeit & Lifestyle	(Leisure & Life-style)	771 (19.78%)
Gesellschaft & Politik	(Society & Politics)	872 (22.37%)
Immobilien & Wohnen	(Accommodation & Living)	805 (20.65%)

in the hold all category *other* while only 1.5% of the examples are classified as *staff* pages, which makes this dataset particularly difficult to classify.

This dataset had been collected earlier [3], but we still had to discover its hyperlink graph by canonizing URLs and identifying all those that point into the dataset. Consequently, not all predecessors of a page are present in this dataset, only those that have been collected in the dataset. As a result, its graph structure is dramatically weaker connected than that of Allesklar. No predecessor could be found for 5082 pages of the dataset and only 67 pages have more than 10 predecessors. The connectivity of the two datasets is shown in Figure 2.

5.2 Classifier

As base classifier for the text classification experiments we used SVM-light in V6.01 [7]. For handling multiple classes we used two different binarization schemes: *round-robin* or *pairwise* classification, and a weighted version of *one-against-all* where a separate classifier is learned for each problem "class i against all $c-1$ classes other than i". $c-1$ votes are given to class i if the classifier predicts i, 1 vote is given to all other classes if the binary classifier does not predict i. Empirically, this seemed to work quite well.

Different feature types can be combined in two different ways:

Merging: Features of different sources are pooled together.
Tagging: Features of different sources are treated separately.

Thus, if the word "word" occurs in two different feature sources (e.g., in the anchor text and the preceding heading) it is treated as the same feature (and counted twice) in the merging approach, and it is treated as two different features

Table 3. Class distribution for the WebKB dataset

Category	8264 Examples
student	1639 (19.83%)
faculty	1121 (13.56%)
course	926 (11.21%)
project	506 (6.12%)
department	181 (2.19%)
staff	135 (1.63%)
other	3756 (45.45%)

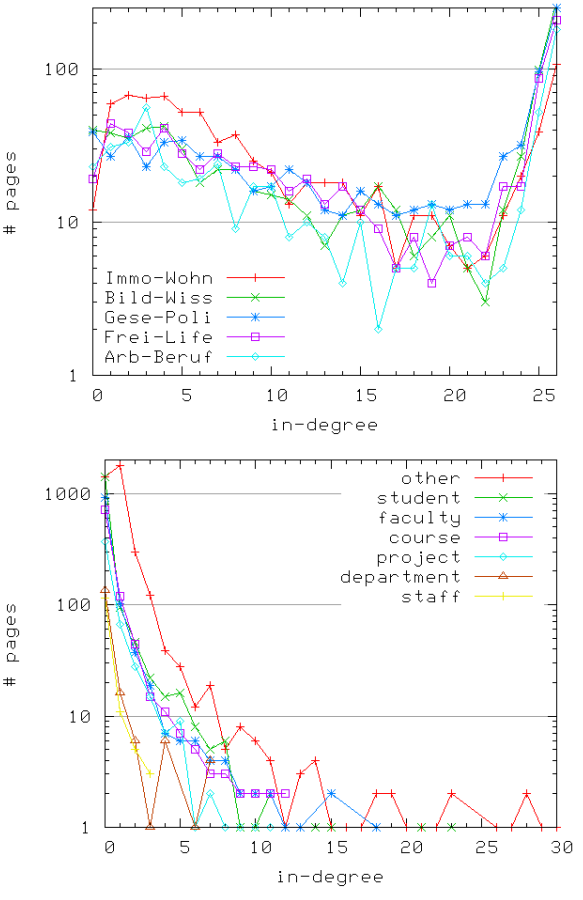


Fig. 2. Inlink-distribution of the documents of the Allesklar and Web→KB datasets

(denoted as "Link Description:word" and "Link Headings:word") in the tagging approach.

As each page has up to 25 preceding pages, we need a technique for combining the features of the predecessors. We compared three different techniques for that:

Meta-Predecessor: The features of all predecessors are pooled together into a single document, called the *meta-predecessor*. This is basically identical to the first approach evaluated by [2], except that we do not pool all features of the predecessor documents but only those extracted by our feature extraction techniques.

Hyperlink Ensembles: Each predecessor is treated as a separate training document (labeled with class label of the page it points to). Predictions are made for each hyperlink separately and combined via voting (cf. Section 3 and [6]).

Mixed Approach: This combines both approaches: Training is like in Meta-predecessor, but for prediction, the trained classifier is used on each hyperlink separately, as in the Hyperlink Ensembles approach.

For all evaluations we used 10-fold stratified cross-validation. We measured recall, precision and the F_1 measure. We report macro-averaged results. As the classes are all of similar sizes, micro-averaged results are quite similar and can be found in [12].

6 Results

Unless mentioned otherwise, we use the following settings in the subsequent sections: the breadth of the *Link Neighborhood* feature set has been fixed to 20 words: 10 words before the anchor and 10 words after the anchor, the text of the anchor is excluded. We use *one-against-all* binarization for handling the multi-class problem, a *meta-predecessor* for combining the features of different predecessor pages, and features from multiple extraction patterns are *merged* together.

6.1 Comparing Different Feature Types

We first evaluate each feature extraction pattern in isolation (Tables 4 and 5). The two first lines represent the macro-averaged precision and the recall for the six classes, and the third line shows the F_1-value computed from these averages. The last two lines show the number of documents in the dataset that were covered by the feature pattern among 3898 documents, i.e., where at least one feature was detected with the respective method, and the number of different features extracted.

A few interesting observations can be made from this table. Although the pattern that covers the most examples is *Own Text*, it is not the pattern that derives the highest number of features. Both, *Link Neighborhood* and *Link Paragraph* retrieve a higher number of features.

For Allesklar, even though they are not applicable to all examples, they dominate a conventional text classifier in both, recall and precision. *Link Neighborhood* increases the F_1 value by 30 in comparison to *Own Text*. The other features seem

Table 4. Results for single feature patterns on the Web→KB dataset

	Link Description	Link Paragraph	Link Neighborhood	Link Headings	Link List Headings	Own Text
Precision	35.54%	29.17%	41.07%	28.35%	17.38%	45.37%
Recall	21.35%	16.71%	17.94%	17.37%	14.89%	24.71%
F1	26.68	21.25	24.97	21.54	16.04	31.99
Coverage	1,143	2,715	3,006	2,828	1,644	8,276
# Features	2,594	51,831	14,360	13,070	4,319	12,658

to be less useful, at least on their own. In particular the features derived from the headings produce very low recall and are also not very precise.

For Web→KB, the results are not so good. Here, each of the link-based features covers only considerably less than half of the examples, and hence their performance is inferior to that of the *Own Text* classifier. It should, however, be noted that this is primarily due to the low number of predecessor pages that were included in the dataset. Also, the rather strict definition of the pattern extractors may have contributed to the bad performance. In [6], the experimental setup was basically the same except that the feature extractors were defined more loosely and heuristically, which resulted in a higher coverage of individual features, and hence in a better overall performance.

6.2 Neighborhood of an Anchor

In the previous section, by far the best results were obtained with the use of features in the neighborhood of the links. However, the notion of neighboring words is vague and is computed with a parameter. In order to get an idea about the sensitivity of this parameter, we computed the macro-averaged F_1-score for each possible combination of 0 to 30 words before the anchor and 0 to 30 words after the anchor text. The anchor text itself (*Link Description*) is also included in these experiments.

The results for the Allesklar dataset show a continuous growth of the F_1-score (Figure 3). Before 20 words, the precision increases quickly. After 20 words, the precision still increases, but very slowly, while the dimensionality (the complexity of the classification problem) still grows. The best compromise for the scope of the neighborhood is around 20, which we distribute equally before and after the anchor (10 words before the anchor and 10 words after).

In the Allesklar dataset we did not observe a decrease of the F_1-score with growing size of the neighborhood in the parameter range that we had tried. This would suggest that using the full text of the predecessor pages would be the best strategy. However, the results of Chakrabarti et al. [2] suggest the opposite: in their experiments (on different datasets) this approach did not result in improved performance.

Our results on the Web→KB data are somewhat less reliable, but despite the high variance, we can observe that there is a clear ridge near the "10 words before" point, suggesting that choosing too large a neighborhood is futile.

Table 5. Results for single feature patterns on the Allesklar dataset

	Link Description	Link Paragraph	Link Neighborhood	Link Headings	Link List Headings	Own Text
Precision	80.00%	79.15%	84.65%	71.80%	70.18%	71.67%
Recall	43.48%	34.30%	67.30%	29.33%	26.66%	32.17%
F1	56.34	47.86	74.98	41.65	38.64	44.41
Coverage	3,653	2,715	3,664	2,672	1,870	3,831
# Features	4,211	79,588	41,513	32,832	4,118	37,898

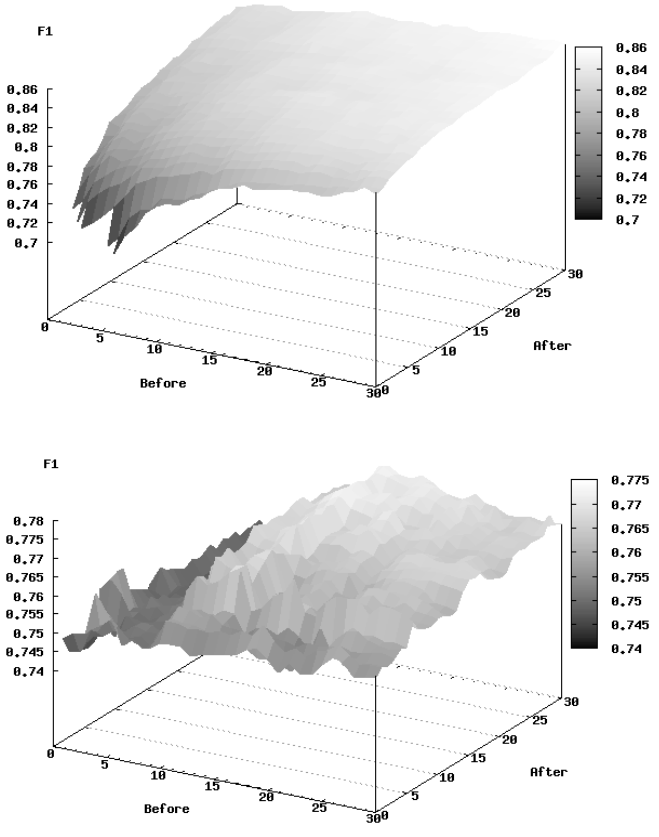


Fig. 3. F1-score for Allesklar (left) and Web→KB (right) for *Link Neighborhood* with different numbers of words before and after the anchor text

However, our experiments on this matter are obviously not yet conclusive, and need to be refined in future work.

6.3 Pairwise Combination of Features

In the previous sections we have seen that some of the features derived from predecessor pages are already sufficient to outperform a conventional text classifier. In this section we investigate whether we can further improve upon these results by combining the extracted features via *merging*.

In Table 6, we summarize these results for the Allesklar dataset. The diagonal of the table (with a dark gray background) shows the macro-averaged precision π and the F_1-score of each individual feature set (the other values can be found in Table 5). The upper triangle (light gray background) shows the recall (ρ) and precision (π) results for each combination of features in the respective lines and columns, the lower triangle (white background) shows their coverage c and the F_1-score.

Table 6. Results of pairwise combinations of features. The upper-right triangle shows precision (π) and recall (ρ), the lower-left shows coverage and the F_1-score.

	Link Neighbor.	Link Description	Link List Headings	Link Headings	Link Paragraph	Own Text
Link Neighbor.	π=84.65% F_1=74.98	π=**84.89%** ρ=65.67%	π=**84.87%** ρ=67.31%	π=84.15% ρ=63.8%	π=82.72% ρ=58.88%	π=82.58% ρ=58.44%
Link Description	c=3678 F_1=74.05	π=80.00% F_1=56.34	π=**80.01%** ρ=42.15%	π=76.68% ρ=38.5%	π=76.44% ρ=36.19%	π=75.75% ρ=37.1%
Link List Headings	c=3665 F_1=**75.08**	c=3653 F_1=55.21	π=70.18% F_1=38.64	π=**71.83%** ρ=28.78%	π=**79.66%** ρ=26.77%	π=**72.36%** ρ=33.82%
Link Headings	c=3665 F_1=72.58	c=3653 F_1=51.26	c=2744 F_1=41.09	π=71.80% F_1=41.65	π=70.09% ρ=26.62%	π=**72.34%** ρ=35.11%
Link Paragraph	c=3667 F_1=68.79	c=3655 F_1=49.12	c=3013 F_1=40.07	c=3103 F_1=38.59	π=79.15% F_1=47.86	π=72.51% ρ=34.87%
Own Text	c=3898 F_1=68.44	c=3898 F_1=49.81	c=3864 F_1=**46.10**	c=3879 F_1=**47.28**	c=3882 F_1=47.09	π=71.67% F_1=44.41

In **bold** font, we show the combinations that outperform both of its constituent patterns in terms of precision (upper triangle) or F_1 (lower triangle). It can be seen that, although using two feature types instead of one does not always increase the performance, it may yield a substantial improvement in some cases. For example, the headings of a preceding list improve the precision in all combinations with other features. However, this gain has to be paid with a loss in recall. In terms of the F_1-score, not many improvements are observable. Heading-based features occassionally make a difference, but these improvements are not likely to be of significance (except when combined with the original text).

The detailed results for Web→KB can be found in [12]. They were qualitatively similar, as can also be seen from the ranking of the individual features (Tables 4 and 5). The best-performing combination was *Link Description* with *Link Paragraph*, which had a precision of 56.66% at a recall of 20.35%.

In summary, these results show that combining these feature sets may be helpful, and even feature types that, on their own, do not yield good results, may be very useful in combination with other types.

6.4 Different Classification Methods

In this section, we study the influence of the choice between the options discussed in Section 5.2, namely between *meta predecessor*, *hyperlink ensembles*, and the *mixed approach*, between the binarization algorithms *one-against-all* or *round-robin* and between the feature combiners *merging* or *tagging*. In total, there are 12 possible combinations of these options, resulting in 12 different methods.

We ran the classification process for the 6 feature sources available and for the 15 combinations of two of those feature sources. Detailed results can be found in [12]. For a summary statistic, we report the average rank (1–12, in terms of

Table 7. Ranking of the different methods for Allesklar

Combination	Binarization	Non-local	Avg. Rank	Avg. π	Avg. ρ
Merging	One against all	Meta predecessor	1.14	78.37%	43.76%
Tagging	One against all	Meta predecessor	1.86	77.35%	42.25%
Merging	One against all	Hyperlink Ensembles	2.86	73.17%	33.42%
Tagging	One against all	Hyperlink Ensembles	3.38	72.43%	32.51%
Merging	One against all	Mixed Approach	5.57	68.98%	37.77%
Tagging	One against all	Mixed Approach	5.95	67.85%	36.61%
Merging	Round Robin	Meta predecessor	6.43	66.32%	59.51%
Tagging	Round Robin	Meta predecessor	7.00	64.95%	57.95%
Merging	Round Robin	Hyperlink Ensembles	8.14	61.64%	48.36%
Tagging	Round Robin	Hyperlink Ensembles	9.10	59.83%	47.50%
Merging	Round Robin	Mixed Approach	10.57	57.71%	50.44%
Tagging	Round Robin	Mixed Approach	10.86	56.00%	48.62%

precision), and the average precision and average recall for each of the methods for each of those 21 atomic experiments on the Allesklar dataset (Table 7).

In this domain, the observed pattern is very regular: our version of *one-against-all* consistently outperformed *pairwise classification*, *merging* consistently outperformed *tagging*, and the *meta predecessor* consistently outperformed the *hyperlink ensembles* and the *mixed approach*. The results in the Web→KB domain were similar but not as consistent [12].

7 Conclusions

The most important result of our study is that using features from predecessor documents may result in a largely improved classification performance, thereby shedding new light on the negative results of [2]. The key difference of our approach is that we suggest to focus on the part of the predecessor texts that is in the neighborhood of the outgoing link, whereas [2] use the complete text of neighboring pages. In our experiments, this technique worked best when used with the simplest approach, namely to merge the extracted features into a single predecessor document. However, we could not compare our approach to all competitive approaches. For example, we have not yet performed a direct comparison to approaches based on the neighbors' class distribution [2,9], which, however, we think are not as general as our technique. We also have not yet tried the full set of extracted features, only pairwise comparisons. Our feature extraction methods could also form the basis of multi-view learning algorithms [11], and it would be interesting to find out whether the advantage of multi-view learning over single-view learning over the union of the views carries over to this scenario as well. Finally, we note that the problem of extracting useful features in the neighborhood of outgoing links is quite similar to approaches that formulate information extraction as a classification problem, for which local features are derived around the item to be extracted.

References

1. S. Brin and L. Page: The anatomy of a large-scale hypertextual Web search engine. *Computer Networks* (1998) 30(1–7):107–117. Proceedings of the 7th International World Wide Web Conference (WWW-7), Brisbane, Australia.
2. S. Chakrabarti, B. Dom, and P. Indyk: Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM SIGMOD International Conference on Management on Data*. ACM Press, Seattle WA (1998), pp. 307–318.
3. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery: Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence* (2000) 118(1-2):69–114
4. T. G. Dietterich: Ensemble methods in machine learning. In J. Kittler and F. Roli (eds.) *Proccedings of the First International Workshop on Multiple Classifier Systems*. Springer-Verlag (2000), p. 1.
5. J. Fürnkranz: Web Mining. In O. Maimon and L. Rokach (eds.) *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag (2005), pp. 137–142.
6. J. Fürnkranz: Hyperlink ensembles: A case study in hypertext classification. *Information Fusion* (2002) 3(4):299–312. Special Issue on Fusion of Multiple Classifiers.
7. T. Joachims: Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol (eds.) *Proceedings of 10th European Conference on Machine Learning (ECML-98)*. Springer-Verlag, Chemnitz Germany (1998), pp. 137–142.
8. J. M. Kleinberg: Authoritative sources in a hyperlinked environment. *Journal of the ACM* (1999) 46(5):604–632. ISSN 0004-5411.
9. Q. Lu and L. Getoor: Link-based classification. In *Proceedings of the International Conference on Machine Learning (ICML-03)* (2003), pp. 496–503.
10. O. A. McBryan: GENVL and WWWW: Tools for taming the Web. In *Proceedings of the 1st World-Wide Web Conference (WWW-1)*. Elsevier, Geneva Switzerland (1994), pp. 58–67.
11. S. Rüping and T. Scheffer (eds.): *Proceedings of the ICML-05 Workshop on Learning With Multiple Views*. Bonn Germany (2005)
12. H. Utard: *Hypertext classification.* Master's thesis, TU Darmstadt, Knowledge Engineering Group (2005)

Information Retrieval in Trust-Enhanced Document Networks

Klaus Stein and Claudia Hess

Laboratory for Semantic Information Technology
Bamberg University
{klaus.stein, claudia.hess}@wiai.uni-bamberg.de

Abstract. To fight the problem of information overload in huge information sources like large document repositories, e. g. citeseer, or internet websites you need a selection criterion: some kind of ranking is required. Ranking methods like PageRank analyze the structure of the document reference network. However, these rankings do not distinguish different reference semantics. We enhance these rankings by incorporating information of a second layer: the author trust network to improve ranking quality and to enable personalized selections.

1 Introduction

The amount of information accessible for everyone is increasing rapidly, mainly driven by computer mediated communication technologies, namely the internet respectively the www. New websites appear, messages are posted, blogs written, papers published etc. No one is able to keep an overview of all these information sources or to find interesting information by herself, so search engines are an important tool to fight the problem of information overload, to "bring order to the web", as the google-programmers Brin and Page [1] call it.

A search engine carries out three important tasks to do its job:[1]

Information gathering: It crawls the web to collect as much of its content as possible, building a huge repository.

Information selection: For each search query it selects the subset of corresponding webpages (e. g. webpages containing a given keyword). For common search terms this subset may contain up to many million websites,[2] so a third step is needed.

Information ranking: The matching webpages are sorted by some ranking and only the highest ranked pages are presented to the user.

[1] In praxis the described steps are highly interdependent, the data structures build up in step one are optimized for fast access in step two and three, and selection and ranking may be done in one step.

[2] For instance, querying google for "Christmas" gives 45 400 000 pages and even "ontology" gives 4 390 000 pages (http://www.google.de/ at July 23, 2005).

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 65–81, 2006.

We focus on the task of information ranking. We will use link analysis to rank documents from a set of selected documents. The prominence (the "visibility") of documents such as websites or scientific papers is calculated based on an analysis of the structure of the underlying document collection, in our case the web. In contrast to content-based analysis, web structure mining analyzes references (links or citations) between documents and computes the visibility of a document based on this link analysis.

The basic idea behind these measures is that a webpage, a scientific paper, a journal etc. will be important if cited by many other important pages/papers/journals. Thus its visibility depends on the number of references and the visibility of the referencing documents. This seams feasible (and works well as the success of google shows) but has one important drawback: each reference contributes in the same way to the visibility of the referred page regardless of *why* it is set. The semantics of the reference is ignored: a scientific paper may cite another one because the author finds it useful and supports the work, but also if the author disagrees and wants to give an opposite point of view.[3] For a human reader these different kinds of links lead to different gradings of the cited document which is not resembled by simple link structure based algorithms.

The differentiation of the semantics of a link is highly important in cases in which the respective document is considered by some or even most users as untrustworthy. Distrust might arise because users belong to different scientific communities and push their own approaches, criticizing at the same time the approaches of the rivaling community as inappropriate. However, distrust can also be expressed from an "official" side as in cases of scientific misconduct: investigations by universities can prove scientific fraud. A recent example illustrates this. In the end of 2005 and beginning of 2006, the South Corean stem-cell researcher Woo Suk Hwang was accused of scientific fraud in two landmark papers published in the journal Science.[4] Investigations by the Seoul National University proved both papers as based on fabricated data whereas his other papers such as the Nature paper dealing with the clone hound Snuppy were considered as valid. Clearly, the fraudulent papers cannot any longer be cited by a scientist in a positive way, i.e., confirming the validity of Hwang's work. However, there will be in the next months or even years citations to Hwang's papers in publications dealing with scientific fraud. These links obviously do not declare Hwang's work as valid. Nevertheless, citation-based measures would count these new links when calculating the visibility of the papers and still rank them highly. We therefore claim that the semantics of the links has to be considered in visibility measures.

[3] On the web links of disagreement are seldom set because due to the way search engines rank pages each reference set to a page increases its rank which is not what the disagreeing author normally wants (in context of the discussion around link spam in guestbooks a (controversial) new link attribute `rel="nofollow"` was introduced by Google, MSN Search, Yahoo! and others to mark links not to be counted by ranking algorithms, which also could be used for disagreement links in future).

[4] For an overview of the events see, for example, news@nature.com: *Timeline of a controversy*, including references to further information, http://www.nature.com/news/2005/051219/full/051219-3.html, last access April 04, 2006.

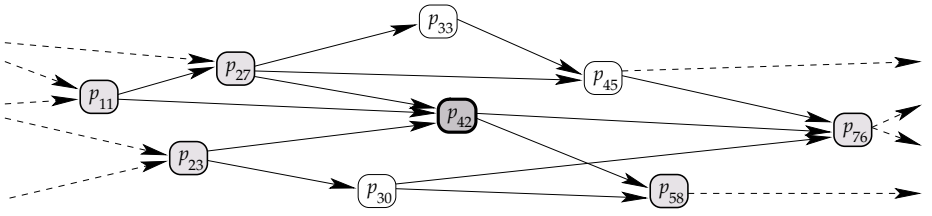


Fig. 1. PageRank example: The visibility vis_{42} of document p_{42} is computed using the visibilities vis_{11}, vis_{23} and vis_{27} of p_{11}, p_{23} and p_{27}: $\text{vis}_{42} = \text{vf}_{p_{42}}(p_{11}, p_{23}, p_{27})$, and vis_{42} itself contributes to vis_{58} and vis_{76}. p_{11} and p_{23} only count half because they also contribute to p_{27} and p_{30} resp. while p_{27} counts one-third for also contributing to p_{33} and p_{45}.

While the semantics of a reference is fairly obvious for the reader of a paper, it is not accessible for a search engine (which simply does not *understand* the text). We therefore propose to reincorporate link semantics to some degree by using a second resource: an author trust network. Additionally this allows for personalizing rankings based on the trust relationships indicated by the user.

We present a framework for extending classical citation-based measures to trust-enhanced visibility functions. Section 2 and 3 introduce the two basic components, namely the document reference network and the author trust network, respectively. Section 4 presents a new approach to capture the reference semantics and to compute the adapted visibility by joining the information from the document and the author trust network. Section 6 summarizes our work and highlights areas for future research.

2 Document Reference Network Based Ranking

One important measure (besides content-based ratings) to rank webpages or other resources referencing each other is to use the link structure to determine the visibility of a certain document. Specifying the structural importance of a page within a document network is a well known problem in social network theory (for an overview see [2,3,4,5]).[5]

In 1976, Pinski and Narin [6] computed the importance (rank) r_a of a scientific journal p_a by using the weighted sum of the ranks r_k of the journals p_k with papers citing p_a. A slightly modified version of this algorithm (the PageRank algorithm) is used by the search engine google [7,1] to calculate the visibility of webpages (with vis_a the visibility of a webpage/document p_a):

$$\text{vis}_a \quad = \quad (1-\alpha) \quad + \quad \alpha \sum_{p_k \in R_a} \frac{\text{vis}_k}{|C_k|}$$

where R_a is the set of pages citing p_a and C_k is the set of pages cited by p_k. Therefore each page p_k contributes by $\frac{\text{vis}_k}{|C_k|}$ to the visibility of p_a (see Fig. 1), and

[5] From a mathematical point of view a document reference network simply is a directed graph with documents (webpages, papers, ...) as vertices and references (links, citations) as edges.

the visibility vis_a of each page p_a is the combination of a base visibility $(1-\alpha)$ and a variable part with contribution $\alpha \in [0,1]$ depending on the visibilities of the papers citing it.

For n pages this gives a linear system of n equations. Solving this equation system is possible but (for large n) very expensive, so an iterative approach is used. All vis_i are set to some default value and then the new values r'_i are calculated repeatedly until all vis_i converge.[6]

The PageRank algorithm works best for networks with cyclic reference structures (links between webpages, citations between journals). For mostly acyclic structures like citations in scientific papers (where documents are temporally ordered and citations go backward in time) similar but slightly different measures are used, which may additionally incorporate metadata like documents age (see e. g. [8]).

All these measures have in common that they are based on the link structure between the documents and blind regarding the "meaning" of a certain reference. We therefore extend citation-based visibility measures to trust-enhanced visibility functions by incorporating link-specific information. The algorithms shown in the next sections work with (mostly) any link structure based ranking algorithm.

3 A Second Source of Information: The Author Trust Network

The information extracted from the document reference network is enhanced with information from a second source of information, the trust network between the authors of the documents. Authors are connected to others via trust statements indicating that the source of the trust statement has a certain degree of trust in the capabilities of the target to provide 'good' documents, for instance to write excellent scientific papers with a well-elaborated argumentation and a 'sensible' opinion from the source's point of view.

Trust statements range from blind trust to absolute distrust represented as numerical values $t \in [t_{\min}, t_{\max}]$, usually $t \in [-1,1]$. Based on direct trust statements between authors, trust relationships can be interpolated between authors who are indirectly connected. This reflects human behavior in so far as we trust to some extent the friends of our friends. A number of trust metrics has been proposed such as the path algebraic trust metric by Golbeck et al. [9] or the spreading activation strategy by Ziegler and Lausen [10]. Most trust metrics are limited to trust values between 0 (no trust), and 1 (maximum trust). Guha et al. [11] discuss research issues on distrust also concerning the development of a metric that is able to cope with trust and distrust. In the following sections we assume to have a fully propagated author trust network.[7]

[6] For a discussion of convergence problems in leaves see [1].

[7] The algorithms described in the next sections simply *use* the trust edges regardless of *how* they were assembled and which algorithm was used to do the propagation (as long as it gives a consistent network). They therefore certainly also work on unpropagated trust networks, although this does not include all information that could be inferred.

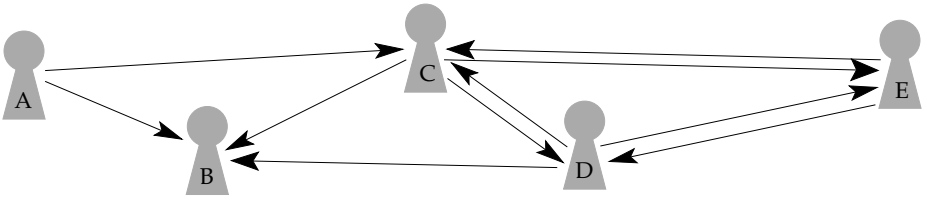


Fig. 2. Author trust network

Trust networks, a specific type of a social network, can be found more and more on the web. Users publish their profiles and indicate whom they know—and trust—in many web-based social communities such as friendster or orkut. These communities had in 2005 already tens of millions members. Social network data has also attracted much attention in the last years as basis for recommender systems in research as well as in commercial applications. Well-known examples are epinions, a product review platform, which uses the explicit trust network between its users to select the product reviews to be presented to a user and ebay's reputation system. The Friend-Of-A-Friend (FOAF) vocabulary has been extended to include trust statements[8] by Golbeck et al. [9]. Users can express in the FOAF files, provided for example at their personal homepages, not only whom they consider as friends but their degree of trust, too. However, to our best knowledge, there is not yet any explicit trust network between the authors of scientific papers available on the web. The simplest approach to build up a basic trust network would be to take directly coauthorships and to derive the weights based on some other information such as the frequency of the coauthorship. Another hint to a relationship is membership in the same institute / organization. Approaches have therefore been presented such as [12] who aim to build trust networks by mining the web. [12] have constructed a trust network for an academic society, namely the Japanese Society of Artificial Intelligence, on the basis of information published in the web. Besides of coauthorship and co-membership in the same laboratory, they consider co-membership in projects and committees and participating in the same conference as indicators for a social relationship. Such network could be used in the trust-enhanced visibility measure. Theoretical foundations and research projects on trust-based recommender systems are provided for instance by Guha [13], Montaner et al. [14] and Kinateder and Rothermel [15].

An author trust network allows to add "meaning" to references: if we know that author A trusts author B in the sense that A likes B's point of view and the quality of her work, we can assume that a reference in a paper of A to a paper of B supports this paper. On the other hand, if an author C distrusts an author D (e. g. D is a creationist and C a darwinist), a reference in a paper of C to one of D normally will not support D's paper. This certainly will not hold for all references but works well enough to improve network reference based rankings.

[8] See the ontology for trust ratings at http://trust.mindswap.org/ont/trust.owl

The author trust network provides additional information which is not deducible from the document reference network. Considering information from an author trust network, it can be distinguished whether a document has a high rank due to its usefulness or because it is very controversially discussed.

4 Trust-Based Visibility: A Two-Layer Approach

Figure 3 shows a two-layered network with the author trust network at the top and the document reference network at the bottom.[9] In the document network information is located in the vertices (the documents' visibilities) while in the trust network information is located in the edges (with $t_{A\to B}$ the value of the trust edge from A to B).

An author has written one to several papers and each paper is written by one to several authors, so the relation "author of" connects both graphs. In the following sections we show how to bring the trust information down to the document reference network and how to modify a visibility function to incorporate this information to get enhanced measures.

4.1 Propagating Trust to the Document Reference Network

In the first step, the trust values are propagated to the document reference network. As its edges are not annotated the idea is to map the trust information from the trust network edges to the document network references. This is done by identifying each document with its author(s) and attributing each reference with the corresponding trust value (e. g. the edge $e_{11\to 42}$ from p_{11} to p_{42} is attributed with $t_{A\to B}$ for A and B being the authors of the referencing respective referenced document). Coauthorship maps more than one trust value to a reference.

Now the visibility of a document p_a can be calculated depending on the visibility of the documents referring to p_a *and* on the trust attributes of these references using a trust-enhanced visibility function vf^t:

$$\mathrm{vis}_a = \mathrm{vf}^t_{p_a}(R_a, E_a) \qquad \text{with } E_a = \{e_{x\to a} \mid p_x \in R_a\}$$

with R_a the set of documents referencing p_a and E_a the set of attributed edges from R_a to p_a. As the next sections show, there are different ways to model how these trust edges contribute to vf^t.

4.2 Associating Weights to Document References

The next step to build trust-enhanced visibility functions is to turn the *attributed* into *weighted* edges. These weighted edges will be used to create the enhanced visibility functions. A visibility measure as described in Sec. 2 handles all references equally. The idea of using weighted references is that the weights modulate

[9] To simplify the example we show an acyclic part of a document network. All algorithms will work on arbitrary graphs, e. g. webpages.

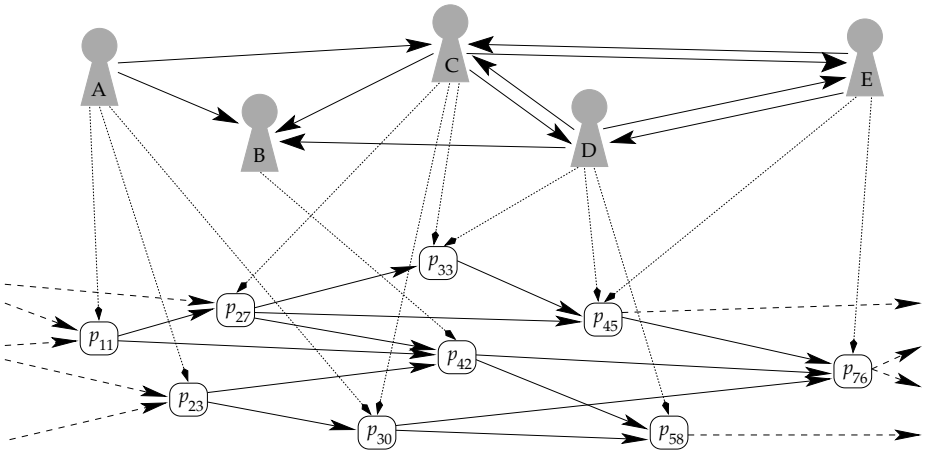


Fig. 3. Combining trust and document reference network

the support in visibility one paper gives to another: if the edge $e_{a\to b}$ from document p_a citing document p_b has a high weight, then the visibility vis_a of p_a will give high support to the visibility vis_b of p_b. In contrast, with a low weight, vis_b is less supported by vis_a.

Depending on the trust network an edge $e_{i\to j}$ may have zero, one or more than one trust value associated. We conflate these values to one value $\bar{e}_{i\to j}$ representing the average trust value. In most scenarios we can assume that distrusting authors will not write a paper together and therefore the attributed trust values do not differ too much, so using the average trust value is feasible.[10] If the edge $e_{i\to j}$ is not attributed by any trust value a fixed value $\bar{e}_{\text{default}} \in [t_{\min}, t_{\max}]$ is used.

The resulting $\bar{e}_{i\to j}$ cannot be used directly as edge weight because trust values may be negative[11] while we need non-negative edge weights.[12] So edge weights $w_{i\to j}$ are computed from $\bar{e}_{i\to j}$ by a mapping function

$$\begin{aligned} I : [-1, 1] &\to [0, m] \\ w_{i\to j} &= I(\bar{e}_{i\to j}) \end{aligned}$$

[10] In other scenarios edges with highly differing trust values (i. e. coautorship of distrusting authors) have to be coped with in a special way.

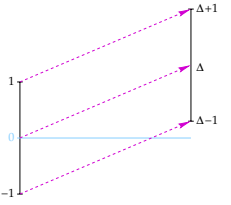
[11] At least in some trust metrics (the interesting ones), trust values can be negative, e.g., $t \in [-1, 1]$, with -1 for distrust and 1 for trust.

[12] Technically many visibility functions can cope with negative weights. Nevertheless the semantics of negative weights is not clear. A visible paper p_i citing p_j with a negative weight would lower vis_j (even to values < 0). This is not reasonable, because each additional citation increases the chances of p_j to be found, even if the citation is deprecatory.

Table 1. Examples for mapping functions

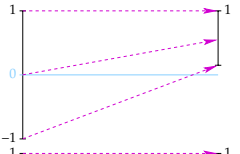
I_+: $w_{i\to j} = \Delta + \bar{e}_{i\to j}$, with $\Delta \geq -t_{\min}$

guarantees non-negative weights, but weights may get values greater 1.



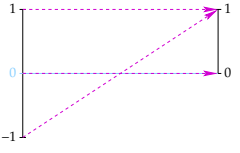
I'_+: $w_{i\to j} = \dfrac{\Delta + \bar{e}_{i\to j}}{\Delta + t_{\max}}$, with $\Delta \geq -t_{\min}$

guarantees $w_{i\to j} \in [0, 1]$.



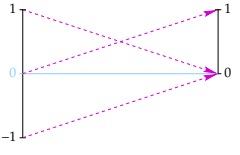
$I_{||}$: $w_{i\to j} = |\bar{e}_{i\to j}|$

highly trusting and highly distrusting references give the same (high) weight.



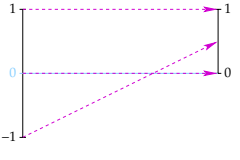
$I_\updownarrow$: $w_{i\to j} = 1 - |\bar{e}_{i\to j}|$

neutral references give most, highly trusting and distrusting references small support.



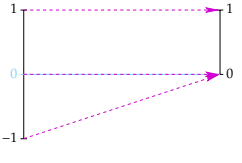
I_λ: $w_{i\to j} = \begin{cases} \bar{e}_{i\to j} & \text{for } \bar{e}_{i\to j} \geq 0 \\ -\lambda \bar{e}_{i\to j} \quad (\lambda \in (0,1)) & \text{otherwise} \end{cases}$

reduces the influence of distrust references (e. g. with $\lambda = 0.5$ they only contribute half).



I_0: $w_{i\to j} = \max\{0, \bar{e}_{i\to j}\}$

distrust references give zero weights.



ensuring non-negative weights. We will see that for some visibility functions edge weights must be between zero and one (i. e. $m = 1$; $w_{i\to j} \in [0, 1]$) while others work with any non-negative weights.

By choosing different mapping functions, different trust semantics can be established. So one can decide how the trust values influence the weights and whether the impact of negative edges (distrust) should be small or large (see table 1): to which amount does a document referencing another document of a trusted author, a distrusted author or a neutral author support its visibility? For example with I'_+ nearly no support is given to distrusted documents while choosing $I_{||}$ the support of highly trusting *and* distrusting documents is high while neutral citations give no support, inversly to $I_\updownarrow$.

Most of the mapping functions I shown give weights spanning the whole interval $[0, 1]$. Modifying I to

$$I_\beta(\bar{e}_{i\to j}) = (1-\beta) + \beta \cdot I(\bar{e}_{i\to j})$$

allows to finetune the influence of the author trust network on document visibility by changing β: for $\beta = 1$ is $I_\beta(\bar{e}_{i\to j}) = I(\bar{e}_{i\to j})$, i. e. maximum impact of the trust network, decreasing β gives decreasing impact and for $\beta = 0$ is $I_\beta(\bar{e}_{i\to j}) = 1$, i. e. no impact of the trust network.

Choosing the right mapping function is an application specific design decision. For some application scenarios it might even be appropriate to consider only distrust links although this seems very strange in the first place. However, analyzing a case of scientific misconduct such as the case of Hwang, a visibility function giving the distrusted papers can support users in the analysis of gray literature remaining after the investigations in cases of scientific misconduct. In these investigations, not every document is clearly proven to be valid or faked, but for a large number of documents by the accused author or his coauthors, validity remains unclear. Highly ranking distrusted documents therefore reflects the community's suspicions about a certain gray paper.

4.3 Weighted Visibility Functions

Now these weights can be used to create trust-enhanced visibility functions. In the most simple case, weights are directly used to modulate any reference network based visibility function making it trust-aware by multiplying the visibility contributed by $e_{i\to j}$ from p_i to p_j by $w_{i\to j}$. For example with PageRank we get the "simple weighted PageRank":

$$\mathrm{vis}_a \quad = \quad (1-\alpha) \quad + \quad \alpha \sum_{p_k \in R_a} \frac{w_{k\to a} \cdot \mathrm{vis}_k}{|C_k|}$$

This lowers the average visibility distributed (for $t_{\min}, t_{\max} \in [-1, 1]$), except for I_+.[13] Therefore introducing weighted references in a certain visibility function may disturb convergence. Renormalization of all visibilities may hence be necessary in each iteration step. By choosing a slightly different approach this can be avoided, as the next section shows.

4.4 Weighted PageRank

According to the PageRank, a page p_r referencing k other pages p_{r_1} to p_{r_k} contributes with $\frac{\mathrm{vis}_r}{k}$ to each of the referenced pages. By modulating the contribution by edge weights $w_{r\to r_i}$, we get the contribution[14]

$$\mathrm{vis}_{p_r \to p_{r_i}} = \frac{w_{r\to r_i}}{\sum_{p_{r_j} \in C_r} w_{r\to r_j}} \mathrm{vis}_r \,.$$

[13] Calculating the visibility without weighted edges is equal to setting all edge weights to 1. So with a weight $w_{k\to a} < 1$ the contribution of p_k to the visibility vis_a of page p_a is lowered from $\frac{\mathrm{vis}_k}{|C_k|}$ to $w_{k\to a} \cdot \frac{\mathrm{vis}_k}{|C_k|}$.

[14] Note that even for $w_{r\to r_i} > 1$ the fraction is below 1, so I_+ can be used as weighting function.

Inserting this into PageRank[15] we obtain the trust-aware visibility function[16]

$$\begin{aligned} \mathrm{vis}_a &= (1-\alpha) + \alpha \sum_{p_k \in R_a} \mathrm{vis}_{p_k \to p_a} \\ &= (1-\alpha) + \alpha \sum_{p_k \in R_a} \frac{w_{k\to a}}{\sum_{p_j \in C_k} w_{k\to j}} \mathrm{vis}_k \end{aligned}$$

This does not change the amount of visibility distributed from one page, but now some references gain more and others less. So this is a kind of local normalization on each page. It is sensitive to relative trust differences, but not to absolute values: the support of a paper by an author A trusting B and distrusting C to papers of B and C is clearly distinguished from the support of a similar paper by an author A' distrusting B and trusting C, but the papers of an author trusting all others and an author distrusting all others give similar support and here the algorithm is not sensitive at all. And at least for a page with only one outgoing reference nothing changes. So as long as the relative trust relations are of interest the approach is feasible, otherwise another function (like the "simple weighted PageRank") has to be used (see section 4.6).

The way and the strength to which trust values contribute to the visibility of a document can be customized by changing the mapping function. Note that this algorithm ensures that the same amount of visibility is distributed as for original PageRank, albeit the amount for single edges change.

Using I_+ as mapping function, the influence of trust can be finetuned by changing Δ: for $\Delta \to -t_{\min}$ only references with high trust contribute while with $\Delta \to \infty$ we get the original PageRank.

$$\text{with } I_+ : \frac{w_{a\to a_i}}{\sum_{p_{a_j} \in C_a} w_{a\to a_j}} = \frac{\Delta + \bar{e}_{a\to a_i}}{k\Delta + \sum_{p_{a_j} \in C_a} \bar{e}_{a\to a_j}}$$

This definition of vf^t supports pages important within a certain community.[17] A page gaining many references from within the authors' community (giving high trust values) raises visibility while a page referenced from outside (low trust values, distrust) decreases visibility. This may be feasible to some account but will not fit all users needs. One may claim: "the best ranked papers are those with only supporting references, but for me controversial papers are of greater interest. And anyway, I want to get the important papers of *my* community, not of others."

This motivates to consider two further aspects in the adapted visibility functions. On the one hand, a trust-enhanced visibility function that favors controversially discussed documents is required. On the other, a personalization of the visibility function permits to better match the users' individual needs.

[15] This modification is not restricted to PageRank, other visibility functions are adaptable accordingly.

[16] To increase efficiency, the fraction $\frac{w_{k\to a}}{\sum_{p_j \in C_k} w_{k\to j}}$ should be precalculated once for all references of the whole net.

[17] A community is a set of authors trusting each other.

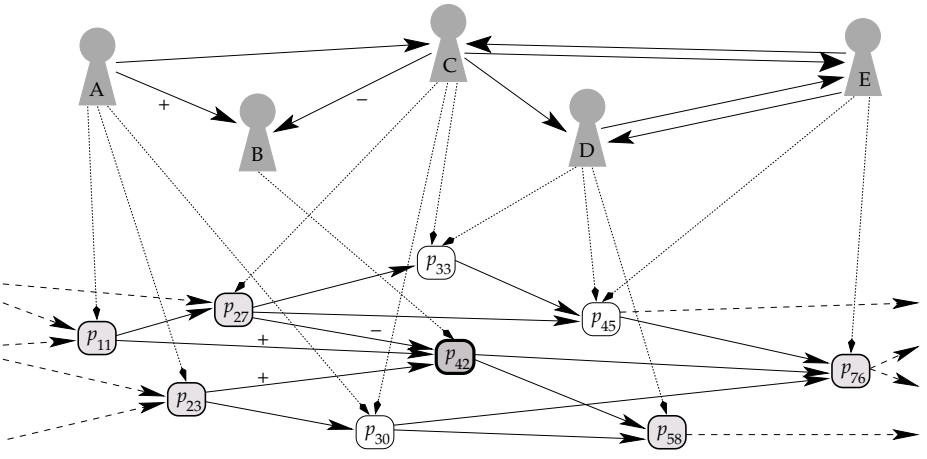


Fig. 4. Controversial documents. Author A trusts B and C distrusts B, therefore p_{11} and p_{23} support p_{42} while p_{27} refuses p_{42}.

4.5 Controversial References

A document is controversially discussed, if it is cited by both supporting and refusing documents. For a given document p_a examining the associated average trust edges $\{\bar{e}_{i\to a} \mid p_i \in R_{p_a}\}$ tells, *how* controversial it is. So (we assume trust values $t_{\min} = -1, t_{\max} = 1$)[18]

$$\widehat{\delta e}_a = \frac{\sum_{p_i,p_j\in R_a} (\bar{e}_{i\to a} \; - \; \bar{e}_{j\to a})^2 \, \mathrm{vis}_i \, \mathrm{vis}_j}{\left(\sum_{p_i\in R_a} vis_i\right)^2}$$

is a measure for the degree of controversy of a document p_a. It provides the highest ranks for the most controversially referenced documents.

Now this measure is used to modify the visibility vis_a of a page p_a, we get

$$\widehat{\mathrm{vis}}_a \quad = \quad \gamma \; \widehat{\delta e}_a \; + \; (1-\gamma) \; \mathrm{vis}_a \; .$$

By choosing γ accordingly the contribution of the degree of controversy $\widehat{\delta e}_a$ and the page visibility, respectively, can be set.

Applying this to any trust-enhanced visibility function gives a measure sensitive to controversially discussed papers. For example with weighted PageRank we get:

$$\mathrm{vis}_a \quad = \quad (1-\alpha) \; + \; \alpha \left(\gamma \; \widehat{\delta e}_a \; + \; (1-\gamma) \left(\sum_{p_k\in R_a} \frac{w_{k\to a}}{\sum_{p_j\in C_k} w_{k\to j}} \, \mathrm{vis}_k \right) \right) .$$

[18] Without distrust ($t_{\min} = 0$) we would not have any controversially discussed papers.

Using this extended formula not only modifies the visibility of a document by its degree of controversy but additionally propagates the modified visibility to the cited documents. In other words: a paper cited by many controversial papers gains visibility. This does not follow the strict definition of controversy as given before and therefore may be an undesired effect. To only support strictly controversial documents the visibility calculation is done in two seperated steps: the visibility of all documents is computed by some arbitrary visibility function, and then for each document p_a its controversy-enhanced visibility $\widehat{\text{vis}}_a$ is computed by the given formula without any propagation.

One inaccuracy is left in this model of controversy because we just used average values $\bar{e}_{i\to j}$ for $\widehat{\delta e}$, which would not give a controversy for a referencing document written by two (or) more coauthors with very different trust values regarding the author of the referenced document. You can claim that we simply do not know why the authors decided to set this reference, therefore taking the average is feasible (as done above). Alternatively the discrepancy can be modeled by changing $\widehat{\delta e}$ to not using the average $\bar{e}_{i\to j}$ but all single trust values attributed to $e_{i\to j}$.

4.6 Personalized Trust Visibility Rankings

The visibility measure described in the last section points out controversial papers but does not incorporate a personal view. An additional step allows for personalizing the rankings: assume we have edges $\bar{e}_{i\to k}$, $\bar{e}_{j\to l}$ of high trust with A being author of p_i and B author of p_j. A user U trusting A and distrusting B will not agree that both edges count equal. A reference from a personally distrusted author simply should count less. So we modify an edge $\bar{e}_{i\to k}$ by $t_{U\to A}$:

$$\bar{e}'_{i\to k} \quad = \quad \frac{t_{U\to A} - t_{\min} + b}{t_{\max} - t_{\min} + b}\, \bar{e}_{i\to k} \qquad \text{with } b \geq 0$$

Now the influence of each reference directly depends on the trust of the user U in the referencing author A (the degree of this dependence is moderated by b with maximum influence for $b = 0$).

These personalized edges substitute the weighted edges from Sec. 4.2. Personalized edges can be used in most trust-aware visibility functions vf^t, e. g. the simple weighted PageRank algorithm (Sec. 4.3), but not the weighted PageRank of (Sec. 4.4) for here the local normalization obliterates the whole effect: as all outgoing edges of one page are multiplied by the same factor (the fraction does only depend on $t_{U\to A}$ (with A the author of the page), and as the local normalization of the weighted PageRank function is only sensitive to the relative relations of the trust edges the weights are not changed.

Note that the described personalization only affects papers *cited by* the papers of a trusted author and not the visibility of the trusted author's papers. This mostly is not the desired effect as normally not only papers cited by a trusted author should be ranked higher than papers cited by a distrusted author but also the papers of a trusted author should be ranked higher than the papers of a distrusted author. This can easily be achieved by modulating the visibility of

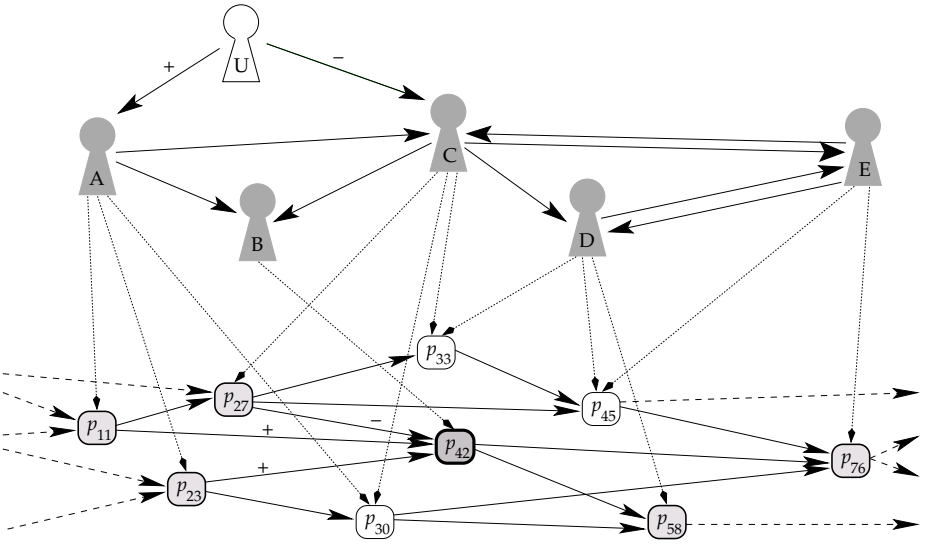


Fig. 5. Personalized Rankings. User U trusts author A and distrusts author C. Therefore from U's point of view references set by A should be considered more important than ones by C.

a paper by the user's trust in its author,[19] which also has impact on the cited papers by propagation. So the formula given here is only an additional tool to increase this effect.

5 Simulation

To show how the trust information from an author trust network can improve visibility measures on a document reference network we evaluate our approach on three simulated scenarios. Each simulation was run 10 times on 10 independent document reference networks with ≈ 3500 documents from ≈ 100 authors each citing 2 to 7 other documents.[20] The mapping function used in all scenarios was I'_+ (with $\Delta = 1$), and the basic visibility function was PageRank (with $\alpha = 0.85$).

The first scenario resembles the situation described in the introduction: an author A is caught lying (writing papers with fake data) and gets untrusted by the others. The document network is set up with 101 auhors trusting each other with $t_{X \to Y} = 1$ (this is similar to using PageRank without weights). The visibility of each document is computed by (weighted) PageRank and the documents are sorted by their visibility. The position of each document in this sorted list gives its rank relative to the others. Now the scientific fraud is detected, and 80

[19] We do not discuss how to do this in detail here for we focus on weighted edges.

[20] Whenever possible an author cites in any new document at least one older document written by herself.

Table 2. The three tables show the results of 10 simulation runs (on 10 different document reference networks) of each of the three scenarios described in section 5. Each row gives the average change in the ranking of the observed documents in percent, the last row gives the average over all runs. Each scenario has its own set of 10 document reference networks (as the author distribution for each scenario is different).

scientific fraud

	$\Delta\%$
1	22.17
2	19.74
3	25.36
4	21.70
5	19.32
6	23.22
7	23.99
8	17.09
9	24.76
10	18.50
avg.	21.59

controversials

	$\Delta\%$
1	14.52
2	13.51
3	15.11
4	13.33
5	16.78
6	13.49
7	14.51
8	15.29
9	15.97
10	12.65
avg.	14.52

personalization

	$\Delta\%$
1	1.55
2	1.94
3	2.05
4	1.79
5	1.95
6	1.60
7	1.62
8	1.91
9	1.73
10	2.27
avg.	1.84

of the 100 other authors suddenly distrust the cheater A (changing their trust to distrust: $t_{X \to A} = -1$, as the validity of his other publications is doubtful). The visibility of all documents is recalculated and the documents are resorted. Now the position of every paper written by A is compared to its position before.[21] In average each paper of A was ranked down by 22% ($s = 0.027$).[22] This means that a paper of A was ranked at position 100 of 3500 (there were only 99 papers with higher visibility), it is now ranked at a position around 870 (there are 869 papers with higher visibility).[23] This shows that the change in trust has an appreciable impact.

The second scenario tests the impact of controversy. The documents are written by three groups of authors: two small groups $\mathcal{A}$ and $\mathcal{B}$ (with 5 authors each) and a large group $\mathcal{C}$ (with 90 authors)). Any author fully trusts the members of its own group ($\forall X, Y \in \mathcal{X} : t_{X \to Y} = 1$). Authors of $\mathcal{A}$, $\mathcal{B}$ trust authors of $\mathcal{C}$ and vice versa ($\forall X \in (\mathcal{A} \cup \mathcal{B}), C \in \mathcal{C} : t_{X \to C} = t_{C \to X} = 0.5$), and authors of $\mathcal{A}$ and $\mathcal{B}$ totally distrust each other ($\forall A \in \mathcal{A}, B \in \mathcal{B} : t_{A \to B} = t_{B \to A} = -1$). So there are two small controversial groups of authors, and papers cited by both groups should gain visibility by using a cotroversy-aware visibility function. Now the visibility of all documents is computed (1) by using weighted PageRank and

[21] We compare the relative rankings of the documents and not the absolute visibility values, as we want to know whether the documents of A are ranked down relative to the others.

[22] The value of 22% is the average of 10 simulation runs on 10 different document reference networks, $s = 0.027$ is the standard deviation.

[23] This need not be true for each single paper of A, some may ranked down more, some less, but for the average of all papers of A.

(2) by using the controversy-enhanced weighted PageRank (with $\gamma = 0.5$) described in section 4.5, and the documents are sorted accordingly. Compared to ranking (1) the controversial documents (i. e. documents cited by at least one author of group $\mathcal{A}$ and one author of group $\mathcal{B}$) are ranked 15% ($s = 0.012$) better in ranking (2),[24] so the controversy-aware visibility function gives the desired effect.

The third setup analyzes the effect of personalization. The documents are written by 101 authors trusting each other ($t_{X \to Y} = 0.5$). A single user U trusts one author A ($t_{U \to A} = 1$) while being neutral to the other 100 authors ($t_{U \to X} = 0$). Now the visibility of all documents is computed using the simple weighted PageRank[25] without (1) and with (2) applying the edge modification of section 4.6 (with $b = 0$). Comparing the document rankings of (1) and (2) shows that documents cited by A are ranked 1.8% ($s = 0.002$) better[26] with personalized edges. Please remember that edge personalization only effects the documents cited by a trusted author but not the documents written by her (which would be accomplished by directly modifying document visibility). The visibility of each document depends on the visibilities of the documents citing it and on the edge weights of these citations. In the given scenario only 1 of 101 authors is considered trustworthy, so for a document cited by A in average this only affects one of the incoming edges of this document (as the other citations are from other authors). Therefore the shift of 1.8% is considerable large.

All scenarios show that using trust information of an author network to introduce edge weights in the document reference network to be able to compute weighted visibility rankings works and gives the expected results.

6 Conclusion and Outlook

In the paper we presented a framework for extending visibility functions to include trust information. The structural information from the document reference network which serves as basis for visibility measures is combined with data from a second source of information, the trust network between the authors of the documents. In contrast to visibility functions as used typically in structural web mining, trust-enhanced visibility functions encompass two novel aspects: on the one hand, they deal with the semantics of the references. A reference can be made due to agreement or disagreement. This is reflected in the proposed visibility functions by considering the trust relationships between authors. We proposed alternative functions which permit requesting users to obtain a ranking that corresponds to their information need: papers which are widely agreed on could be favored by a trust-enhanced visibility function or controversially referenced ones by an alternative one. On the other, integrating trust information permits to personalize the ranking.

[24] Average of 10 document networks.
[25] As discussed in section 4.6 the weighted PageRank is not useful with personalization.
[26] Average of 10 document networks.

An author trust network is not the only possible source of trust information. In another approach (described in [16]) we show how to integrate a trust network between document readers giving recommendations on certain papers. Here the trust information is included in the form of trust-weighted reviews as an additional component in the visibility functions. That is, the reader's opinions on a paper is weighted with the trust the requesting user has in this reader's capabilities to review papers. Depending on the degree of trust in the reviewer, the trust-weighted review determines the new, trust-enhanced visibility or the classical visibility function predominates the result.

Having addressed the basic theoretical foundations of trust-enhanced visibility functions, the described functions were evaluated by a simulation study with three scenarios[27]. In future work evaluation with real data will complement the simulation. The main problem on real world evaluation is to get a trust network built up independently from a document network. As document network for instance the citeseer document collection could be used.

Another important step in future work is to have a closer look on the author trust network. Currently, trust edges are simply projected from the trust to the document network and $\widehat{\delta e}_a$ measures differences in trust weighted references to p_a. In a next step trust edges between referencing authors should be directly taken into account: if author A with paper p_i and author B with paper p_j both cite p_k of author C also the trust edges from A to B and vice versa are interesting. A trusting C *and* B trusting C is a more important sign if A and B *distrust* each other. Evaluating all possible triades of authors (trust/distrust edges) in the author trust network is work in process.

References

1. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
2. Watts, D.J.: Six Degrees: The Science of a Connected Age. W. W. Norton & Company, Inc. (2003)
3. Barabási, A.L.: Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life. PLUME (2003)
4. Menczer, F.: Growing and navigating the small world web by local content. In: Proceedings of the National Academy of Sciences of the United States of America. Volume 99. (2002) 14014–14019
5. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
6. Pinski, G., Narin, F.: Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. Information Processing & Management **12** (1976) 297–312
7. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems **30** (1998) 107–117

[27] Using the COM/TE framework [8], http://www.kinf.wiai.uni-bamberg.de/COM/.

8. Malsch, T., Schlieder, C., Kiefer, P., Lübcke, M., Perschke, R., Schmitt, M., Stein, K.: Communication between process and structure: Modelling and simulating message-reference-networks with COM/TE. JASSS (2005) accepted.
9. Golbeck, J., Parsia, B., Hendler, J.: Trust networks on the semantic web. In: Proceedings of Cooperative Intelligent Agents, Helsinki, Finland (2003)
10. Ziegler, C.N., Lausen, G.: Spreading activation models for trust propagation. In: Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service, Taipei, Taiwan, IEEE Computer Society Press (2004)
11. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, New York, NY, USA, ACM Press (2004) 403–412
12. Matsuo, Y., Tomobe, H., Hasida, K., Ishizuk, M.: Finding social network for trust calculation. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI2004). (2004) 510–514
13. Guha, R.: Open rating systems. Technical report, Stanford Knowledge Systems Laboratory, Stanford, CA, USA (2003)
14. Montaner, M., López, B., Lluís de la Rosa, J.: Opinion-based filtering through trust. In Ossowski, S., Shehory, O., eds.: Proceedings of the Sixth International Workshop on Cooperative Information Agents. Volume 2446 of LNAI., Madrid, Spain, Springer Verlag (2002) 188–196
15. Kinateder, M., Rothermel, K.: Architecture and algorithms for a distributed reputation system. In: Proceedings of the First International Conference on Trust Management. Volume 2692., Springer Verlag (2003) 1–16
16. Hess, C., Stein, K., Schlieder, C.: Trust-enhanced visibility for personalized document recommendations. In: Proceedings of the 21st Annual ACM Symposium on Applied Computing. (2006)

Semi-automatic Creation and Maintenance of Web Resources with webTopic*

Nuno F. Escudeiro and Alípio M. Jorge

LIACC, Faculdade de Economia, Universidade do Porto
nfe@isep.ipp.pt,
amjorge@fep.up.pt

Abstract. In this paper we propose a methodology for automatically retrieving document collections from the web on specific topics and for organizing them and keeping them up-to-date over time, according to user specific persistent information needs. The documents collected are organized according to user specifications and are classified partly by the user and partly automatically. A presentation layer enables the exploration of large sets of documents and, simultaneously, monitors and records user interaction with these document collections. The quality of the system is permanently monitored; the system periodically measures and stores the values of its quality parameters. Using this quality log it is possible to maintain the quality of the resources by triggering procedures aimed at correcting or preventing quality degradation.

1 Introduction

Web characteristics, such as dimension and dynamics [17], place many difficulties to users willing to explore it as an information source. Moreover, information retrieved from the Web is typically a large collection of documents. A query in Google for "Artificial Intelligence" gives, today, a list of 95.000.000 results. Organizing this information conveniently improves the efficiency of its exploitation. To take advantage of the value contained in this huge information system there is a need for tools that help people to explore it and to retrieve, organize and analyze relevant information.

The satisfaction of an information need on the Web is usually seen as an ephemeral one-step process of information search (the traditional search engine paradigm). The user is usually not assisted in the subsequent tasks of organizing, analyzing and exploring the answers produced. Vivisimo (http://vivisimo.com) and Tumba! [25] are exceptions where the retrieved documents are automatically (and immediately) clustered according to syntactic similarity, without any input from the user, other than the keywords of the search query itself. We believe that it is also important to give the user the possibility of specifying how he or she requires the retrieved documents to be organized.

Another important aspect is the existence of persistent information needs. This is the case of many professionals, such as scientists, who need frequent updates about

* Supported by the POSC/EIA/58367/2004/Site-o-Matic Project (Fundação Ciência e Tecnologia), FEDER e Programa de Financiamento Plurianual de Unidades de I & D.

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 82 – 102, 2006.

their area of activity. It is also the case of many societies, (professional or other) engaged in constantly providing up-to-date information on a given topic to their members in the form of a web portal. In this case the information is kept as a web resource by a team of editors, who select and edit the published documents. In some cases, the editor and the web end-user, the person who consults the resource, are the same person. Persistent information needs on a specific topic may be answered by tools that keep an eye on the web, automatically searching for documents on that topic and that are able to present them to the end-users as expected by them. Editors have the role of expressing the end-user's needs and preferences.

In this paper we propose *webTOPIC*, a methodology that assists editors in the process of compiling resources on the Web, with the following characteristics:

- allow for the broad specification of any topic, including its ontological structure, by a team of editors;
- enable the effective exploration of large document collections by the end-user;
- maintain quality, as perceived by end-users, at acceptable levels without requiring explicit effort by the end-user or the editor;
- detect and adapt to drift in end-user needs and to changes in information sources.

From the editor's point of view, webTOPIC is a tool for specifying, collecting and organizing document collections that satisfy some specific and persistent information needs of the (end-)users. Once the editor has specified an information need, webTOPIC compiles resources following these specifications. A *resource* is a document collection satisfying a specific information need. We will refer to each specific user information need as a *topic*. Each resource is an instance of a topic.

The user interacts with the methodology during two distinct phases: in the first phase the user defines the topic and specifies its characteristics (editor's role), in the second phase the user explores the resources that are being compiled by the system (end-user's role). The first phase is concentrated on a short period of time. The specification of a particular information need includes, among others, a taxonomy, which describes the ontological structure the user is interested in, and a set of exemplary documents. In the second phase, which occurs while the user maintains interest on the topic, the system follows the evolution on end-user preferences, automatically and incrementally building and keeping resources aligned with end-user current interests.

In the rest of the paper we start, in section 2, by describing and comparing our approach to previous related work. Then, in section 3, we describe the webTOPIC methodology. We refer to its architecture and then describe its most relevant aspects, including the resource acquisition phase, document pre-processing, learning, resource presentation and exploration and resource quality. Section 4 describes the experiments we have conducted to evaluate our semi-supervised document classification method. In section 5 we present our conclusions and directions for future work.

2 Resource Compilers

An automatic resource compiler is a system that, given a topic, seeks and retrieves a list of the most authoritative web documents, as perceived by the system, for that topic [4]. This is a very broad definition, under which many distinct types of systems

may be considered, including, for instance, search engines. In our work we are interested in an automatic resource compiler, which, given a topic, has the responsibility of building and managing a collection of relevant documents in a continuous effort to keep the collection up-to-date.

Many automatic resource compilation systems and methodologies have been proposed in the past, exhibiting many interesting ideas and characteristics.

Letizia (1995) [18] is a user interface agent that assists a user browsing the Web. Somewhat similar to Metiore [2], Letizia also suggests potentially interesting links for the user to follow. Interest in a document is learned through several heuristics that explore user actions, user history and current context. In Letizia the notion of interest is global to the user; it is not conditioned by the user objective.

The **ARC** system (1998) [4], a part of the Clever project, compiles a list of authoritative web resources on any topic. The algorithm has three phases. In the first phase – search and grow – a query is submitted to AltaVista and a root set is constituted with the top 200 returned documents. This root set is expanded by adding direct neighbors (both in-links and out-links); this expansion step is executed twice so the final expanded set has a radius 2 links larger than the initial root set. In the second phase – weighting – the anchor text is extracted from the documents in the expanded set and links are weighted by the relevance of the terms in the anchor text vicinity. In the third phase – iteration and reposting – an iterative process is carried in order to compute authority and hub measures for each document in the expanded set. The documents with the fifteen highest scores of authority are returned to the user as the authority pages to the topic and fifteen highest scores of hub measure as the topic hub pages.

Personal WebWatcher (1999) [21] is a system that observes users behavior, by analysing page requests and learning a user model, and suggests pages potentially interesting to the user. The system operates offline, when learning user models, and at query time, when proposing interesting pages to the user. When a user downloads a web page the system analyses out-links and highlights those that seem interesting given the specific user model. Similar agents may communicate and exchange information on similar users, leveraging particular experiences, through a collaborative or social learning process. The system learns by exploring requested pages: a page requested to the server is considered to be a positive example of user interest and any links not selected are considered to be a negative example. In this way user relevance feedback is obtained without the need to explicitly request it to the user.

Grouper (1999) [30], operated by the University of Washington between 1997 and 2000, was a document clustering interface, that clustered document collections, at run time, as they were returned by HuskySearch meta-search engine (HuskySearch and Grouper ceased their service in 2000). By generating clusters with simple descriptions Grouper provided a way of organizing search results into collections for ease of browsing. Tumba! [25] is another example of a search engine (http:///www.tumba.pt) that organizes output into clusters. NorthernLight (http://www.northernlight.com) is a commercial search engine, mainly oriented to business, that presents many innovative

search capabilities, including the so called custom search folders, which is, again a way to cluster the output of a query.

Personal View Agent, PVA, (2001) [7] is another personalization system that learns user profiles in order to assist them when they search information in the Web. This system organizes documents in a hierarchical structure – the personal view, which is user dependent and dynamic, automatically adapting to changes in user's interest. Relevance feedback is also obtained implicitly, by analyzing information from proxy server log files; in particular, documents whose visit time is larger than 2 minutes are considered positive examples. All specific personal views (hierarchical structures of categories that represent user interests) are derived from the world view, a generalist pre-defined taxonomy used by default as a starting point of user interests. Personal views are updated by merging and splitting nodes (two crucial operators of the system) in the hierarchical structure according to the perceived interest in each node.

Metiore (2001) [2] is a search engine that ranks documents according to user preferences, which are learned from user historical feedback depending on the user objective. Queries might be based on content and also other document attributes, such as title, author and year. The user must define an objective for each search session. User models consist of the documents that the user has classified in previous sessions. Relevance or interest feedback is explicitly required from the user, who can classify each returned document in one of the following categories: "ok", "known", "?", "wrong". By default all documents are classified as normal, standing for not classified.

Thesus (2003) [11] allows for the users to search documents in a previously fetched and classified document collection. In this system documents are classified based on document contents and link semantics; authors claim that in-link semantics might improve document classification. The system includes four components: the document acquisition module, which starts from a set of seed URLs and crawls new documents following hyperlinks that carry specific semantics; the information extraction module, which extracts keywords from incoming hyperlinks from the documents in the collection and maps them to concepts in the predefined ontology; the clustering module, which partitions the document collection into coherent subsets based on keywords and also on the semantic tags associated to documents in the previous phase; and, finally, the query module, which allows for the user to explore the document collection.

WebLearn (2003) [19] is a system that retrieves documents related to a topic, which is specified through a set of keywords, and then automatically identifies a set of salient topics, by analysing the most relevant documents retrieved in response to the user query that describes the topic. The identification of these salient topics is a fully automatic process that does not allow for user interference.

iVia (2003) [23] is an open source virtual library system supporting Infomine (http://infomine.ucr.edu), a scholarly virtual library collection, from University of California, Riverside, of over 26.000 librarian created and 80.000 plus machine created records describing and linking to academic Internet resources. It is a hybrid system that collects and manages resources, starting with an expert-created collection that is augmented by a large collection automatically retrieved from the Web. iVia

automatically crawls and identifies relevant Internet resources through focused crawling [5] and topic distillation approaches.

In the next chapter we will describe our methodology, webTOPIC, which shares some of these functional characteristics, although the mechanisms that are applied to guarantee them are distinct. Like PVA, the resource organization is dynamically adapted to drifts in user information needs. PVA, Letizia and Personal WebWatcher explore implicit relevance feedback based on the actions performed by the user during search sessions. These user actions are mainly related to server requests for web pages and the time between requests, a requested web page is considered to be a positive example. PVA also organizes resources according to a user specific taxonomy. This taxonomy is always derived from an initial static taxonomy, global to all users. PVA does not allow for the explicit definition of topics, each user has its own taxonomy independently of any specific topic. webTOPIC also exhibits characteristics that are not present at any of the previously discussed systems among which we may stress:

- the presentation phase that we consider fundamental to help the user take advantage of his or her resources;
- the definition of corrective and preventive procedures, to be automatically executed in order to keep system performance at acceptable levels, responding to present or predictable drifts in user information needs;
- topic organization defined by the user through examples.

3 webTOPIC Methodology

The webTOPIC methodology executes a continuous loop where each of the iterations consists on the following phases (adapted from [10,16]):

- *Acquisition*: aims to find and retrieve, from the Web, as many relevant documents as possible while retrieving as few non-relevant documents as possible.
- *Pre-processing*: comprises any transformation process that is applied to the retrieved documents to generate appropriate document models.
- *Learning*: intends to find patterns and to learn how to satisfy user needs.
- *Analysis and Presentation*: aims to facilitate the exploration of large document collections and to detect and adapt to drift in user interests.

3.1 Architecture

Our methodology covers these phases as shown in Fig.1.

The first iteration for a new topic goes from topic definition to document archival (tasks 1 through 8). This is a work conducted by the editor and the result is a first version of the resource. From this point on, the process splits, and two distinct threads are executed:

- in the first thread, the user explores the resource using a common browser (task 9). Usage data is collected so that feedback on current user's interests is implicitly supplied. This data is later analyzed (task 10) to detect any change of user's interests,

– in the second thread (automatically scheduled and triggered) the system periodically refreshes the resource by retrieving new (or updated) documents from the web and classifying them (tasks 2, 5, 6 and 8).

Explicit editor effort is required exclusively for topic specification, including exemplary document labeling (tasks 1 and 3). The methodology is responsible for collecting and analyzing end-user behavior, continuously improving the resource's quality.

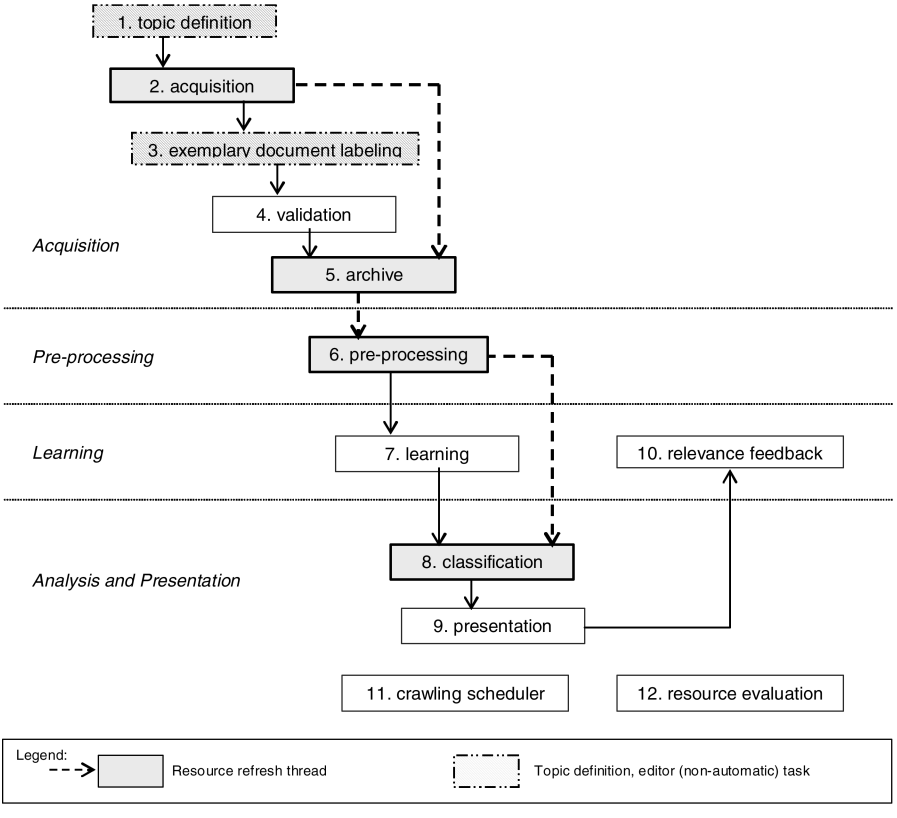


Fig. 1. webTOPIC architecture

3.2 Resource Acquisition

During resource acquisition the system retrieves documents of a given topic and archives them locally – document retrieval is made through a meta-search process by invoking existing web search engines, such as Google or Altavista.. Before that, the editor is required to define the topic of interest. From this specification, the methodology retrieves an initial set of documents that will be used to learn the topic taxonomy. From this point on the methodology will periodically execute the meta-search process in order to refresh the resource.

Topic Specification. Topics are specified by the editor, at tasks 1 and 3 (Fig.1). The most relevant features of the topic specification include:

- a set of representative keywords, which will be used to generate the queries to submit at the meta-search process – for instance "computer" (task 1),
- a taxonomy, representing the ontological structure of the topic – for instance ("hardware", "software", "book") – (task 1) and
- a partially classified set of exemplary documents that should include labeled documents on every taxonomy categories (task 3).

The topic taxonomy is a hierarchy of concepts specifying the ontological structure of the resource. The root is the topic itself. The taxonomy is merely a way of structuring the resource according to user specific needs. Web documents are multi-faceted [3] and it is not possible to know which particular facet the user is interested in unless specified.

Each document in the resource is associated to just one category – the most specific category in the taxonomy adequately representing the document. We assume that there is no uncertainty as to which particular facet is the user interested in and that the topic categories are not ambiguous. Under these assumptions a singular category might be associated to each document without ambiguity.

Primary Data Set Acquisition and Cross-classification. Once a set of keywords is specified, an initial set of documents is retrieved from the Web. The editor is then required to classify a subset of these documents according to the topic taxonomy. We will refer to this document collection, which contains a set of exemplary documents, as the *primary data set*.

Human classification of web pages is highly subjective, inconsistent and erroneous [15, 20]. Minimizing human classification errors at the primary data set is of crucial importance for the accuracy and global performance of the methodology. To minimize human errors, webTOPIC allows performing what we have called *cross-classification*: instead of relying on a single editor, the primary data set may be simultaneously classified by a team of editors. The labels assigned by these independent editors are then merged and ambiguous documents, those that have been assigned distinct labels by distinct editors, are set apart for disambiguation. Each of the editors in this team is assisted in the manual classification task by the manual classifier tool (Fig. 2). This tool reads the URL list file and the topic taxonomy – where it adds the special categories *unlabeled* and *negative* – and allows for the editor to preview and assign categories from the topic taxonomy to documents in the URL list.

Document Retrieval. Meta-search Process. The first instance of a topic, which we have called the primary data set, is the consolidation of the results returned by the first meta-search cycle executed for the topic. Afterwards, the resource is automatically updated from time to time. The dynamic nature of the Web requires special attention so that we always have an up-to-date view of its content. The process in charge of keeping the freshness of the resource, accomplishes the following tasks:

- detect and retrieve new documents, that are not yet part of the resource;
- keep resource documents updated, by retrieving them, analyzing their characteristics and decide whether they have changed since the last retrieval cycle;

– purge broken links; documents that are no longer available online must be excluded from the resource or, at least, marked as broken links.

Whenever a web document is retrieved for the first time, its content changes or it is considered discontinued, the resource log is updated in order to keep a record of the resource evolution.

Fig. 2. Manual classifier

At every meta-search cycle the search engines that have been producing better results for the topic are selected and topic keywords are submitted to each one of them. After submitting the queries to these search engines, the answers are processed to extract URLs into a text file. This consolidated list of URLs, the result of a retrieval cycle for a specific topic, must be classified. If it comes from the primary acquisition step the classification is a manual and partial process – step 3 (Fig. 1); otherwise it is automatic and full – step 8 (Fig. 1).

Resource Instantiation. The instantiation of a new version of the topic – a new resource – depends on the current topic update mode. Two distinct modes are defined:

– *Content*; in this mode a new resource is instantiated whenever the number of accumulated changes – retrieving of a new document, change in the content of document and detection of broken link – that were identified at the resource since its instantiation, reaches a pre-defined threshold. The initial value of this threshold is pre-defined but it is dynamically updated by the system.
– *Time*; in this mode a new resource is activated periodically; the period is defined by the number of retrieval cycles that have been executed since resource activation. The initial value of the activation period is pre-defined but webTOPIC dynamically updates it.

3.3 Document Pre-processing

Once archived, each document is submitted to pre-processing. This phase is responsible for transforming an HTML document into its representation in the selected document modeling framework. It includes a *data preparation* step, which extracts the required set of features from HTML files, and a *data representation* step, which builds the document model from the set of features previously extracted from the document file.

Data Preparation. During data preparation any linguistic or structural symbols, such as HTML tags, are eliminated. Eliminating these symbols from the document reduces the feature set size, and therefore the computational effort. It is also important to identify the language of the document. This is crucial if we want to process multi-lingual resources. Language identification may be done from language profiles [22].

Data Representation. We propose a document model combining two document description levels – content and metadata – that are potentially valuable sources of evidence concerning the classification of HTML documents. Each of these complementary aspects is characterized by a set of features:

- *Content* description level is characterized by text words. Document content is represented in the vector space model. Each document is represented by its TF×IDF vector [1].
- *Metadata* includes first fetch date, current version date, URL, status, status date, number of retrieval cycles without change and automatic and editor labels.

Detecting Document Version and Duplicates. A simple heuristic is applied to detect distinct versions of the same document and document duplicates: when receiving a new document, we compute the cosine between the new document and all other documents in the resource and select the highest cosine value. This value is then used to decide whether the document is new or potentially duplicated, based on a predefined threshold (*copy* threshold). This threshold is, by default, set to 0,9 if there is no previous study supporting a distinct value.

According to this copy threshold, documents are considered:

- *potential duplicates*, if their cosine is above copy threshold,
- *distinct documents*, if their cosine is below copy threshold.

Potential duplicates are further analyzed to decide whether they are true duplicates or distinct versions of the same document. They are considered true duplicates if their cosine equals 1 or if the *term-gap* between them is less than 5% – again, this figure is set by default; otherwise they are considered distinct versions of the same document. The term-gap between two documents is computed from their content – TF vector – as the ratio of the sum of paired differences between term frequencies by the minimum of the sum of term frequencies at each document.

3.4 Learning

Learning the topic taxonomy in an unsupervised manner, by applying clustering techniques, does not seem appropriate. The user may be interested in an organizational

structure different from the one obtained with unsupervised techniques. On the other hand, a supervised learning scheme requires a large number of labeled examples from each category. This is a major drawback since the manual classification of web pages is highly time-consuming.

We use a semi-supervised solution, requiring the editor to classify a few examples from the primary data set at an initial phase. The system will then learn a classifier for each category in the taxonomy, based on the exemplary pre-labeled documents, using semi-supervised techniques. It is only required that the set of pre-labeled examples covers all the taxonomy categories.

Although the document model includes two distinct sets of attributes only its content – TF×IDF vector – is used to learn the taxonomy (from the metadata features, only human labels from the exemplary documents are used). Since we are relying exclusively on content, we will apply standard text classifiers for classification purposes.

Standard Text Classifiers. When applied to web pages, classical text mining methods treat each page independently. These methods explore the page content text, ignoring links between pages and the class distribution of neighbour pages. Several methods are available:

- *Rochio*'s algorithm [12] is a classic method for document categorization in information retrieval. In this method the training examples are used to build a prototype vector for each class. The prototype vector for each class is computed as the average vector over all the training document vectors that belong to the class. A new document is classified according to the distance measured between the document vector and the class prototype vectors.
- *Naive Bayes* methods [12] use the joint probability of words and categories to estimate category probabilities given a document. Dependency between words are ignored, i.e., the method assumes that the conditional probability of a word given a category is independent from the conditional probability of any other word given the same category; this is the naive assumption.
- *k-Nearest-Neighbours* (kNN) is an instance based classifier which has obtained good results in pattern recognition and text categorization problems [26, 28]. This method classifies a document based on the characteristics of its closest *k* neighbours documents. Documents are represented in the traditional vector space model and the similarity measure is the cosine between document vectors. The categories that have a relevance score above a given threshold are assigned to the document [29].
- *Support Vector Machines* (SVM) [13] are based on the intuition that a hyper-plane that is close to several training examples will have a bigger chance of making erroneous decisions than one which is as far as possible from all training examples. The SVM algorithm is a binary classifier that defines a maximum margin hyper-plane between the convex hulls formed by the training examples of each class.

SVM and kNN classifiers are frequently referred to as the most accurate classifiers for text [6]. We have assessed the performance of these two classifiers in experiments described in section 4.2.

Semi-supervised Learning. webTOPIC uses a Support Vector Machine (SVM) classifier [13] – currently the most accurate classifiers for text [6] – wrapped in a simple semi-supervised algorithm called *bootstrapping* [14]. In this method, the classifier is wrapped in a process that iteratively labels unlabeled documents and adds them to the labeled set. This cycle is executed until one of the stopping criteria is met. One of these stopping criteria is based on the concept of classification gradient, which is introduced here as a way of measuring model improvement between iterations.

We have defined *classification gradient* as the percentage of unlabeled documents that have been assigned distinct labels from one iteration to the next. It is computed as the number of documents that have been given, at the current iteration, a different label relatively to the last iteration, divided by the total number of training documents. We assume that if a classifier, generated at a given iteration during the semi-supervised process, does not produce significantly different labels with respect to the previous iteration, the discriminative power of the classifier on the data has already been exhausted.

This is used as one of the five stopping criteria for the semi-supervised classifier:

1. The classification gradient is below the minimum threshold. The discriminative power of the data set has already been captured by the classifier;
2. None of the predictions has a posterior probability greater than a pre-defined threshold. All unlabeled documents are ambiguous to the classifier.
3. Number of errors committed on human labeled documents increases when compared to last iteration. Classifier judgment is diverging from users perspective.
4. Number of iterations is greater than or equal to a pre-defined threshold. This criterion halts the bootstrapping process if it is not converging.
5. The number of unlabeled documents in the training data set is below a given minimum. Tries to avoid potential over-fitting.

The classification algorithm is described in pseudo-code:

initialize *labeled* and *unlabeled* datasets
while (none of the stopping criteria is met) {
    save last generated classification model
    learn classification model from *labeled* set
    compute predictions for *unlabeled* as *new.predictions*
    set *newly.labeled* = *new.predictions* with *posterior* > *min*
    add *newly.labeled* to *labeled* and remove from *unlabeled*
    compute #errors on human labels
    compute classification gradient
    compute *missed.human.label*
    save predictions as *previous.predictions*
    save *previous.human.errors* as #errors on human labels
}
return last saved classification model

At each iteration a SVM classifier for each category in the topic taxonomy is generated and is applied to classify the documents at the unlabeled data set. For each unlabeled document, the algorithm compares *posterior* – the posterior probability of the most probable category – with a pre-defined minimum (*min*). If the category

probability is higher than this pre-defined minimum the predicted label is accepted and the document is marked as newly labeled, meaning that it has been labeled at the current iteration; otherwise we assume that there is not enough evidence to accept the predicted label for that document. When all the unlabeled examples have been processed, the documents marked as newly labeled at the current iteration (*newly.labeled*) are added to the labeled data set and removed from the unlabeled data set. No special attention is given to misclassification of labeled data unless when it occurs at human-labeled documents; the *missed.human.label* variable is set if the current iteration misclassifies more human-labeled documents than the previous one.

When any of the stopping criteria is met, the bootstrapping cycle stops, the current SVM model is discarded and the previous one will be returned and used to classify new documents to be included in the resource.

3.5 Resource Presentation

Presentation Layer. This is implemented as a graphical interface and has two main purposes: it allows the specification of an information need – a topic – and it enables the exploration of the results, the resources. It helps users to explore large resources and to analyze particular documents while bearing in mind the whole collection and the relationships between sets of documents.

This presentation layer provides two distinct views of the topic: the organizational view and the exploratory view.

Organizational View. The organizational view is a simple interface between users and resources oriented for resource manipulation. In this view the resource is seen as a directory tree where each node represents a concept of the topic taxonomy; the root directory stands for the topic itself. Each directory contains the set of documents that were labeled with the corresponding category.

Editors can use this interface to manipulate the resource, for instance, by moving documents from one directory to another if they want to redefine the document's label. Changes in the resource – newly retrieved documents, documents that have changed and broken links – are identified by distinct colors: a newly added document has its name printed in green until the user opens it for the first time, documents that have changed since the last access will be printed in yellow and documents that have become broken links are colored red.

Document icons show the labeling mode – manual or automatic – followed by the posterior probability of the label, when the label has been automatically assigned. Other properties – essentially metadata attributes – are directly available through the document icon.

This interface has a set of functionalities, which allow the editor to reorganize the topic, such as:

- topic taxonomy may be updated by creating new concepts at any node in the taxonomy, moving nodes from one location to another or removing nodes.
- moving a document from one node to another is assumed by the system as explicit editor re-labeling, corresponding to the manual classification of that document.

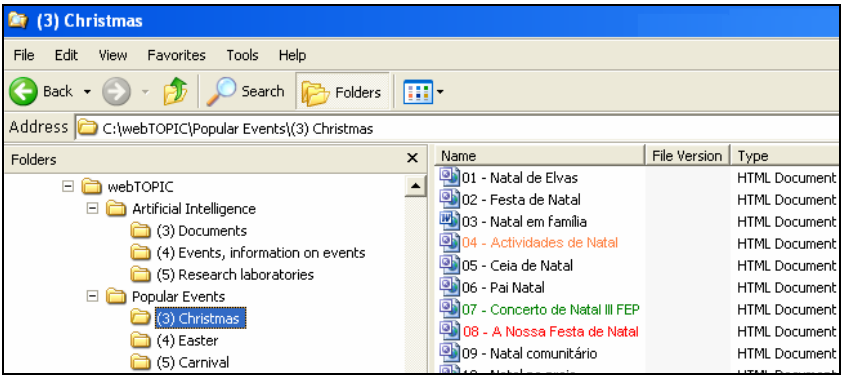


Fig. 3. Resource presentation, organizational view

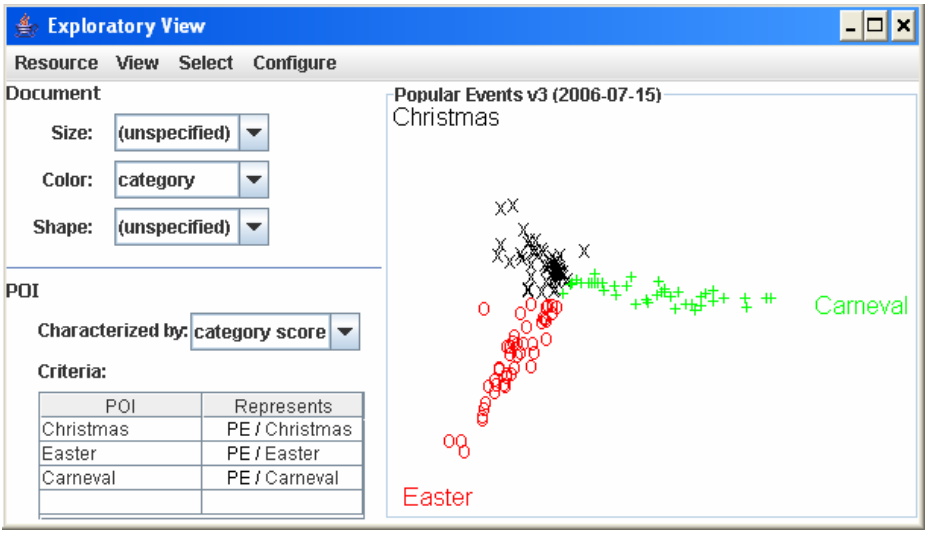


Fig. 4. Resource presentation, exploratory view

Exploratory View. The exploratory view is the graphical interface to the resources. It is meant for users who want to understand the internal structure of the resource and how topic categories are covered and related. This view is inspired in the Vibe system [24] and supports users in the management of the details and structure of large document collections, namely:

– characterization of points of interest (POI) [24] based on several properties from documents. A POI is a concept, of relevance for the topic, characterized by a set of attributes; documents are associated to POI and are shown on the graphical interface according to their similarity based on these attributes;
– size, color and shape of the icons representing documents may be dynamically associated to several features;

- allow the definition of the relative importance of POI;
- eliminate all the selected – or all but the selected – documents from the display;
- view all the documents that share some specific feature, or set of features, with any given document or set of documents.

3.6 Analysis and Resource Quality

Resource Quality. Resource quality is measured on a three-dimensional quality space along the dimensions: *Automation*, *Efficacy* and *Efficiency*. Automation is the complement of the workload that was required to the editor. Efficacy is an aggregation of precision, accuracy and soundness. Efficiency aggregates recall, freshness and novelty [1, 8]. A resource's quality is a value between 0 – lowest quality – and 1 – highest quality. Quality is measured comparatively to an ideal resource, which would have a quality of 1, and would completely satisfy resource users.

All quality factors – workload, precision, accuracy, soundness, recall, freshness and novelty - are periodically and automatically computed, from a set of quality indicators that count the number of documents that present certain characteristics. Workload, for instance, is computed as the ratio of the number of exemplary documents that were manually labeled by the editor by the total number of documents that were obtained at the acquisition phase.

We introduce *soundness* here to measure the validity of a given document that may be accurately classified and fresh but that describes the topic at some level that is not interesting to the user – it might be very superficial, or too technical. The soundness of a document may change as the user gets acquainted with the resource, so it should be measured within a limited period time. Document soundness is a kind of relevance not only regarding the topic itself but also depending on the depth of the knowledge of the end-user. Soundness is measured from the number of recent requests for the document; it is computed as the percentage of document hits relative to the total number of hits on documents of the same category, measured over recent past, say last 100 hits.

Quality indicators that depend on resource usage are inferred from user feedback. User feedback is recorded from relevant user actions – such as move, print and view – performed while exploring resources, without requiring explicit editor effort. These user/resource interaction logs are processed to update documents metadata. Periodically the system computes the required quality indicators and computes and records the resource quality factors and coordinates.

Adaptive Quality Control. Quality indicators, factors and coordinates are computed and recorded by the system itself, while in operation. This way, webTOPIC can continuously evaluate the current position of resources in the quality space and try to correct any eventual degradation on a specific quality factor, dimension or at a global level. Quality values are measured and stored to analyze the evolution and forecast resources quality. Specific corrective procedures are pre-defined to be automatically triggered when the system quality falls off some established limit thresholds. The application of forecasting techniques allows the execution of preventive measures, which are also defined. The methodology computes and records the evolution of the system quality and executes the preventive and corrective procedures, whenever necessary.

The methodology just described is not yet fully supported by the current webTOPIC system. At the present we have a preliminary version that implements tasks 1 to 8

(Fig. 1), which allows specifying, compiling and organizing resources. The presentation, resource evaluation and adaptive quality control features are still being deployed.

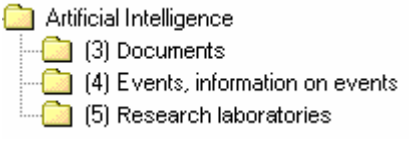
4 Experimental Results on Document Classification

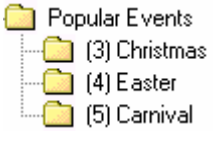
The accurate classification of web pages and, consequently, the correct organization of the resource, based on a small set of just a few manually labeled examples, are crucial to the global performance of the entire system. In this chapter we describe the experimental study that has been conducted with the goal of evaluating the adequacy of semi-supervised learning for the automatic resource compilation setting. We will try to analyze two fundamental aspects of the classification task:

– accuracy of the classifier on its own and
– robustness against human errors during the initial pre-classification task.

4.1 Resources

We have chosen two distinct topics to perform our experiments: *Artificial Intelligence* (AI), which is of relevance for us, and *Popular Events* (PE), which is a topic that easily provides document collections. We are interested in organizing these topics according to the following taxonomies:





The resource on the topic AI has 66 documents and the resource on PE has 200 documents. All documents in both resources are in Portuguese, since we are using a stemming algorithm and a stop-words list for the Portuguese language. This requirement became a major drawback when acquiring resources, especially on the topic AI.

Both resources were fully labeled on their respective taxonomies, observing the following distributions:

Table 1. AI resource

Category	#	%
Documents	22	33,3
Events, information on events	23	34,8
Research laboratories	21	31,8
Artificial Intelligence (total)	66	100,0

Table 2. PE resource

Category	#	%
Christmas	75	37,5
Easter	66	33,0
Carnival	59	29,5
Popular Events (total)	200	100,0

These documents, from both resources, were processed by the webTOPIC prototype generating a set of informational structures. The lexicons of AI and PE resources have 5262 and 3732 terms, respectively. From these, we have excluded terms that appear in just one document. This allows for a significant reduction in the feature space dimension without loosing discriminative power since terms that appear in just one document are irrelevant for the classification task [27]. This way the effective dimensions of document-by-term weight matrices become 200×2240 for PE and 66×1427 for the AI topic.

4.2 Experimental Setup

Since we classify web pages based on content we will apply text classifiers. In a preliminary phase we have chosen SVM and kNN – which usually present good performance at text classification tasks – and decide which, among them, is the best classifier given our datasets. Then we will use the selected classifier in the following experiments and try to evaluate its accuracy and robustness.

SVM and kNN accuracy were evaluated with a 10-fold cross-validation process with both our datasets. The kNN classifier was used with k=5 (five neighbors). The results are summarized in Table 3 that presents the average error rate.

Table 3. Supervised accuracy

	Artificial Intelligence		Popular Events	
	Error rate	Std. dev.	Error rate	Std. dev.
SVM	27%	5,2%	12%	4,1%
kNN	38%	4,0%	24%	2,8%

These experimental results led us to apply SVM. The learning task will be based on the semi-supervised bootstrapping algorithm described in section 0 and on the SVMLight classifier (http://svmlight.joachims.org/).

4.3 Experiments

The accuracy of the SVM classifier was initially estimated for the supervised setting; this served as a reference to be compared with the semi-supervised setting.

To estimate the accuracy of our classifier we have generated several partitions of each of the resources, obtained by random sampling. Each partition divides the resource into two subsets: one of them is used to train and the other one to test the classifier. Training data sets contain a number of documents that ranges from 3 to 60, in the case of AI resource, and from 3 to 180, in the PE resource in multiples of three – the maximum train size is 90% of the resource size. We have generated 10 random samples for each of these training data sets and estimate generalization error rates by the mean computed over these 10 folds.

The minimum error rate, which we will use as our reference, has a value of 25%, for the AI resource – obtained when training with 42 examples and testing with 24 – and a value of 8%, for the PE resource – obtained when training with 45 examples and testing with 155. The results are summarized in Table 4.

This previous study, under supervised conditions, conducted us to assess the semi-supervised setting on data sets with 45 training examples and 155 testing examples, for the PE resource, and with 42 training examples and 24 testing examples, for the AI resource.

Table 4. Supervised accuracy

	Artificial Intelligence	Popular Events
Generalization error	25%	8%
#Training data set	42	45
#Testing data set	24	155

For each experiment the training data will be split in two parts: one where document labels are made available to the classifier and another one where document labels are hidden from the classifier. The number of labels available to the classifier ranges from 3 to 39 (AI resource) or 42 (PE resource).

Training examples were obtained, for each experiment, by two distinct methods: random sampling and stratified sampling. Random samples are the same that have been previously used at the supervised setting. Stratified samples are obtained by

Table 5. Semi-supervised accuracy for AI

#labeled	Error	
	random	stratified
3	0,65	0,65
6	0,61	0,54
9	0,68	0,49
12	0,70	0,47
15	0,70	0,38
18	0,60	0,37
21	0,60	0,28
24	0,46	0,33
27	0,40	0,30
30	0,34	0,25
33	0,32	0,26
36	0,27	0,25
39	0,28	0,25

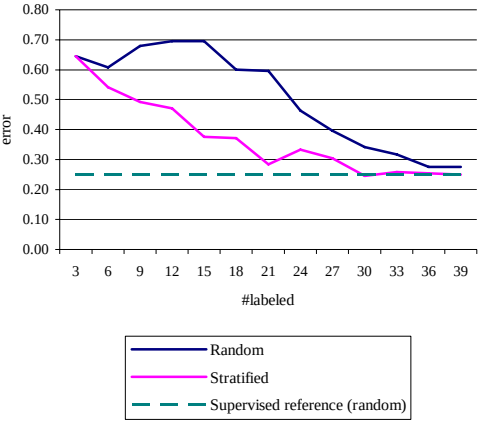
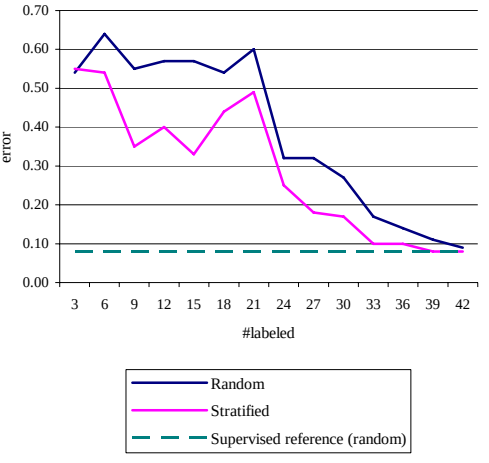


Table 6. Semi-supervised accuracy for PE

#labeled	Error	
	random	stratified
3	0,54	0,55
6	0,64	0,54
9	0,55	0,35
12	0,57	0,40
15	0,57	0,33
18	0,54	0,44
21	0,60	0,49
24	0,32	0,25
27	0,32	0,18
30	0,27	0,17
33	0,17	0,10
36	0,14	0,10
39	0,11	0,08
42	0,09	0,08



merging three random samples (one for each category) with the same number of examples, each one extracted from the set of training documents that have a given category.

Stratified sampling is analyzed because we wish to understand the influence that an asymmetric set of pre-labeled documents might have on the performance of semi-supervised classification, which may suggest special care on this subject at the exemplary document labeling task. Both random and stratified samples were tested and the results are presented at Tables 5 and 6.

From these experiments, we are lead to believe that semi-supervised learning reduces the workload of the resource editor without compromising accuracy. For the PE resource we have achieved, with the supervised setting, a minimum error of 8%, with a workload of 45 labeled documents. Applying semi-supervised learning we have an error rate of 10% for a workload of 33 documents, thus reducing the editor's workload in 27% for nearly the same accuracy.

Special care should be taken to guarantee, as far as possible, that the taxonomy categories are uniformly distributed over the human labeled exemplary documents since unbalanced distributions deteriorate classifiers accuracy.

Robustness was evaluated on data sets derived from the ones used to estimate accuracy. Testing data are the same. Training data sets have the same examples but labels were deliberately corrupted. These errors emulate human classification errors.

Erroneous labels have two properties: the original label where the error was committed and the erroneous label that was assumed instead of the true one. Committed errors may be either random or systematic, concerning both these properties. Random errors are committed in documents from distinct categories while systematic errors are consistently committed on documents of the same category. On Table 7 we refer

to these properties by *Random*, for random errors, or by *SystematicN*, for systematic errors, where *N* stands for the category where the error was committed. We characterize four distinct error profiles based on these two properties and generate four training groups, one for each error profile. We have computed error rates for each error profile with increasing percentage of inserted errors.

The robustness of our classifiers may be evaluated from these results. The resource on PE exhibits an increase of 3,5% – over the original error – at the generalization error rate for an increase of 1% at the training data set, if errors are random. This degradation is worse, as expected, if errors are systematic.

Table 7. Slope of error rate by inserted errors percentage

	Trend line slope	
Error profile	AI resource	PE resource
Random/Random	6,1%	3,5%
Systematic3/Random	6,2%	5,2%
Systematic3/Systematic4	6,3%	5,2%
Systematic3/Systematic5	7,7%	7,2%

5 Conclusion

We have designed an automatic system, aimed at compiling informational resources on the web, which adapts to user information needs and changes in information sources, providing tools that help exploring large resources. Such a system allows individuals and organizations to create and maintain focused web resources.

The classification of web documents is a critical task since the resource quality perceived by the end-user is highly influenced by the classifier accuracy. Moreover, this task is still more critical because training documents are very expensive to obtain and high classification accuracy requires large training data sets. Semi-supervised classification algorithms are particularly suitable under these circumstances.

In our experiments, semi-supervised text classification obtained error rates comparable to the supervised setting, but for lower workloads. We have achieved a reduction of 27% on workload, without significant increase in error rates. This is especially the case if exemplary documents are stratified according to the distribution of labels in the resource. Biased label distributions at the exemplary documents deteriorate classifiers accuracy.

Concerning robustness, experimental evidence indicates that systematic errors – committed on exemplary documents labeling – produce worse effects on error rate than random errors. While random errors introduce white noise that equally affects all labels, systematic errors introduce erroneous patterns that explicitly misguide the classifier.

We have proposed and implemented a methodology that already accomplishes some of the aims mentioned above. In the following we identify some of the current shortcomings of our proposal and suggest paths for improvement. Our prototype uses flat classifiers that ignore inheritance properties and hierarchical relationships

between the classes that constitute the topic taxonomy. Applying hierarchical classification techniques [9] might improve classification.

The application of information extraction techniques may also add very interesting capabilities. These techniques may generate document summaries, or summarize the content of sets of documents, which may be very valuable at the presentation layer. We may explore information extraction techniques to automatically build a report on the current state of the art of some topic, specifically structured according to the organization preferred by the user.

When defining a topic the user may choose any arbitrary taxonomy; webTOPIC does not impose any kind of restriction or rule on this subject. We intend to develop a web document model, consisting of several independent sets of features, covering most of the aspects that the user may explore to organize the resource. The classification task would then proceed in two steps: in the first step it learns the most adequate set of features for the topic and in the second step it uses that set of features to learn the taxonomy. This classification process may improve the flexibility of the topic definition.

References

1. Baeza-Yate, R., Ribeiro-Neto, B. (1999), Modern Information Retrieval, Addison Wesley
2. Bueno, D., David, A.A. (2001), "METIORE: A Personalized Information Retrieval System", Proceedings of the 8th International Conference on User Modeling, Springer-Verlag.
3. Buntine, W., Perttu, S., Tirri, H. (2002), "Building and Maintaining Web Taxonomies", Proceedings of the XML Finland 2002 Conference, pp 54-65.
4. Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., Kleinberg, J. (1998), "Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text", Proceedings of the 7th International World Wide Web Conference.
5. Chakrabarti, S., Berg, M., Dom, B. (1999), "Focused crawling: a new approach to topic-specific resource discovery", Proceedings of the 8th World Wide Web Conference.
6. Chakrabarti, S. (2003), Mining the web, Discovering Knowledge from Hypertext Data, Morgan Kaufmann Publishers.
7. Chen, C.C., Chen, M.C., Sun, Y. (2001), "PVA: A Self-Adaptive Personal View Agent System", Proceedings of the ACM SIGKDD 2001 Conference.
8. Cho, J., Garcia-Molina, H. (2000a), "Synchronizing a database to improve freshness", Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data.
9. Dumais, S., Chen, H. (2000), "Hierarchical Classification of Web Content", Proceedings of the 23rd ACM SIGIR Conference, pp 256-263.
10. Etzioni, O. (1996), "The World-Wide-Web: quagmire or gold mine?", Communications of the ACM, Vol. 39, No. 11, pp 65-68.
11. Halkidi, M., Nguyen, B., Varlamis, I., Vazirgiannis, M. (2003), "Thesus: Organizing Web document collections based on link semantics", The VLDB Journal, 12, pp 320-332.
12. Joachims, T. (1997), "A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization", Proceedings of the 1997 International Conference on Machine Learning.
13. Joachims, T. (1998), Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Research Report of the unit no. VIII(AI), Computer Science Department of the University of Dortmund.

14. Jones, R., McCallum, A., Nigam, K., Riloff, E. (1999), "Bootstrapping for Text Learning Tasks", IJCAI-99 Workshop on Text Mining: Foundation, Techniques and Applications, pp. 52-63.
15. Kobayashi, M., Takeda, K. (2000), "Information Retrieval on the Web", ACM Computing Surveys, 32(2), pp 144-173.
16. Kosala, R., Blockeel, H. (2000), "Web Mining Research: A Survey", SIGKDD Explorations, Vol. 2, No. 1, pp 1-13.
17. Levene, M., and Poulovassilis, A., editors. "Web Dynamics: Adapting to Change in Content, Size, Topology and Use", Springer, 2004.
18. Lieberman, H. (1995), "Letizia: an Agent That Assists Web Browsing", Proceedings of the International Joint Conference on AI.
19. Liu, B., Chin, C.W., Ng, H. T. (2003), "Mining Topic-Specific Concepts and Definitions on the Web", Proceedings of the World Wide Web 2003 Conference.
20. Macskassy, S.A., Banerjee, A., Dovison, B.D., Hirsh, H. (1998), "Human Performance on Clustering Web Pages: a Preliminary Study", Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining.
21. Mladenic, D. (1999), Personal WebWatcher: design and implementation, Technical Report IJS-DP-7472, SI.
22. Martins, B., Silva, M.J. (2002), "Language Identification in Web Pages", Document Engeneering Track of the 20th ACM Symposium on Applied Computing (Unpublished).
23. Mitchell, S., Mooney, M., Mason, J., Paynter, G.W., Ruscheinski, J., Kedzierski, A., Humphreys, K. (2003), "iVia Open Source Virtual Library System", D-Lib Magazine, Vol. 9, No. 1.
24. Olsen, K.A., Korfhage, R.R., Sochats, K.M., Spring, M.B., Williams, J.G. (1992), "Visualization of a Document Collection: The VIBE System.", Information Processing & Management, Vol. 29, No. 1, pp 69-81.
25. Silva, M.J., Martins, B. (2002), "Web Information Retrieval with Result set Clustering", Natural Language and Text Retrieval Workshop at EPIA'03.
26. Yang, Y., Chute, C. G. (1994), "An example-based mapping method for text categorization and retrieval", ACM Transaction on Information Systems, pp. 253-277.
27. Yang, Y., Pederson, J. (1997), "A Comparative Study of Feature Selection in Text Categorization", International Conference on Machine Learning.
28. Yang, Y. (1999), "An Evaluation of Statistical Approaches to Text Categorization", Journal of Information Retrieval, vol. 1, nos. 1/2, pp 67-88.
29. Yang, Y., Slattery, S., Ghani, R. (2002), A Study of Approaches to Hypertext Categorization, Kluwer Academic Publishers, pp. 1-25.
30. Zamir, O., Etzioni, O. (1999), "Grouper: A Dynamic clustering Interface to Web Search Results", Proceedings of the 1999 World Wide Web Conference.

Discovering a Term Taxonomy from Term Similarities Using Principal Component Analysis

Holger Bast[1], Georges Dupret[2], Debapriyo Majumdar[1], and Benjamin Piwowarski[2]

[1] Max-Planck-Institut für Informatik, Saarbrücken
{bast, deb}@mpi-inf.mpg.de
[2] Yahoo! Research Latin America
{gdupret, bpiwowar}@yahoo-inc.com

Abstract. We show that eigenvector decomposition can be used to extract a *term taxonomy* from a given collection of text documents. So far, methods based on eigenvector decomposition, such as latent semantic indexing (LSI) or principal component analysis (PCA), were only known to be useful for extracting *symmetric* relations between terms. We give a precise mathematical criterion for distinguishing between *four kinds of relations* of a given pair of terms of a given collection: unrelated (car - fruit), symmetrically related (car - automobile), asymmetrically related with the first term being more specific than the second (banana - fruit), and asymmetrically related in the other direction (fruit - banana). We give theoretical evidence for the soundness of our criterion, by showing that in a simplified mathematical model the criterion does the apparently right thing. We applied our scheme to the reconstruction of a selected part of the open directory project (ODP) hierarchy, with promising results.

Keywords: Taxonomy Extraction, Ontology Extraction, Semantic Tagging, Latent Semantic Indexing, Principal Component Analysis, Eigenvector Decomposition.

1 Introduction

Eigenvector decomposition has proven to be a powerful tool for a variety of machine learning and information retrieval tasks. Its use comes under a variety of names: principal component analysis (PCA), latent semantic indexing (LSI) or latent semantic analysis (LSA), multidimensional scaling, spectral analysis or spectral learning, and many more. In this introduction, we first explain the common principle behind all these names. We then show how we make novel use of this principle to derive a term taxonomy, given only a collection of text documents with no external knowledge base whatsoever.

For eigenvector decomposition to be applicable, the data must be suitably cast into *matrix* form. For collections of text documents, the following *document-term matrix* representation is standard. Each row corresponds to a document, and each column corresponds to one of the words occurring in the collection (the

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 103–120, 2006.

so-called vocabulary). An entry in the matrix is, in the simplest case, a count of how often the respective word occurs in the respective document. This can be normalized in a number of ways, for example, by having different weights for different terms, or by normalizing the norm of each column of the matrix to be 1. In any case, for each document only a small fraction of the words will have a non-zero entry, so that the matrix is very sparse. This has practical importance because sparse matrices can be decomposed much more efficiently than dense ones: the computational complexity is essentially proportional to the number of non-zero entries. A keyword query can be represented just like a document, with a non-zero entry for each query word.

Given the matrix representation, the similarity between two documents, or between a document and a query, can be measured by the similarity between the corresponding (high-dimensional, yet sparse) row vectors. A typical measure of similarity between two such vectors is the so-called *cosine similarity*, which is just the dot-product between the two vectors, divided by the product of their Euclidean lengths.

With an appropriately normalized document-term matrix, this vector similarity approximates the true similarity of the documents (as perceived by a human with regard to their contents) surprisingly well, except for one principal drawback. If two documents, or a document and a query, have no words in common, their similarity is zero, even if they are about the same topic but just happen to use different words to describe it. For example, there might be two texts on automatic taxonomy extraction, one of them indeed using the word `taxonomy`, but the other consistently using the word `hierarchy`. The similarity of the two vectors will then be relatively low, thus not accurately reflecting the strong similarity in topic.

Eigenvector decomposition of the document-term matrix is a way to overcome this problem. The basic idea is to project the documents as well as the query from their high-dimensional space to a space of a given much lower dimension k. This space is spanned by a set of eigenvectors derived from the document-term matrix. In the simplest case, these are just the eigenvectors pertaining to the k largest eigenvalues of the product of the document-term matrix with its transpose (the so-called term-term correlation matrix). The idea is that while the dimensions in the original space correspond to words — which are different even if they mean similar things — the dimensions in the lower-dimensional space correspond to a set of semantic concepts underlying the collection.

Many variants of this approach have been proposed and shown useful for a wide variety of machine learning and information retrieval tasks. In [8] and [3] it has been shown that for collections of text documents, eigenvector decomposition methods essentially work by identifying *pairs of related terms*.

1.1 Our Contribution

All previous applications of eigenvector decomposition in information retrieval tasks were *symmetric* in the sense that the implicitly identified term relations, in the sense of [3], were symmetric.

In this paper we extend the results of [10, 11] which showed that eigenvector techniques indeed have the power to discover asymmetric relationships between terms, in particular hyponym/hypernym relationships as in `banana - fruit`, which together form a taxonomy of terms. This is the first time eigenvector techniques have been used — and shown to be useful — for such a purpose. In particular, we give a mathematical criterion for determining the relationship of a given word pair: unrelated, symmetrically related, or asymmetrically related. In the case of an asymmetric relation, the criterion identifies a direction, for example, it finds that `banana` is more specific than `fruit` and not vice versa. We show that in a simplified mathematical model our criterion classifies all term pairs correctly.

The criterion does not (and cannot possibly) distinguish between whether the more specific term is a *kind of* the more general term like in banana - fruit (hyponym/hypernym relation), or a *part of* it like in finger - hand (meronym/holonym relation), or just some unspecific *aspect of* it like in disease - outbreak. This restriction holds for all automatic taxonomy extraction schemes that do away completely with an external knowledge base; see the discussion of related work in the following section.

We tested our schemes on two hierarchies derived from the open directory project (ODP) and manually checked the quality of the relations that were found. The results are promising. There is no standardized benchmark for assessing a given taxonomy with respect to a given corpus. In previous works, two sources of quality assessment were provided. The first were extensive examples. The second where small user studies, asking users to assess whether a set of given relations makes sense. The examples were usually more expressive in terms of how the method works than the user studies. In this paper, we give only examples.

1.2 Related Work

The work closest in spirit to ours is by Sanderson and Croft [25]. They say that term A *subsumes* term B if and only if the set of documents containing term A is (approximately) a superset of the set of documents containing term B. This condition is checked for each pair of terms from a pre-selected set. Like in our approach, no knowledge whatsoever on what the terms mean is used, that is, each term could be replaced by a unique number, with no change in result. As is common in vector-space based approaches, even the positions of the terms in the text are ignored, that is, if each document were sorted alphabetically (which would effectively make it unreadable for a human) exactly the same term taxonomy would be extracted.

According to [3], our eigenvector decomposition approach in this paper can be seen as way to assess the relations of all term pairs without the need for actually looking at each term pair explicitly. In particular, we have no need for an explicit pre-selection of terms as in [25].

The work by Sanderson and Croft has been extended in a number of ways. For example, Nanas *et al* [22] first identified symmetric relations between term pairs (via occurrence counts in a fixed-size sliding window) and then made those

relations with a large difference in (a sort of) document frequency between the two terms asymmetric. Glover *et al* [14] have explored the use of *anchor texts* in a hyperlinked environment to determine the level of hierarchy of a term; they do not explicitly compute individual relations though. Joho *et al* [17] showed the usefulness of the subsumption hierarchies from [25] for interactive query expansion.

Much work in information retrieval is concerned with building so-called *topic hierarchies*, that is, finding a hierarchy of (relatively few) topics that describe the collection well, and for each topic providing a succinct summary. The first task is usually achieved by some form of hierarchical clustering. Summaries are provided by giving few descriptive terms. Lawrie and Croft have shown good results with this approach using language modelling techniques, for the topic generation as well as for the summarization [20] [19] [18]. Chuang and Chien have shown web search snippets to be useful [4]. A survey of the large body of work along these lines is beyond the scope of this paper. Note that eigenvector decomposition methods perform a kind of *soft clustering* on the collection, in the sense that each document is assigned not to one but to a number of topics (implied by the few selected eigenvectors).

It is surprising and interesting that fully automatic methods, oblivious of any meaning of the words and sometimes, like ours, even of their position in the text, can contribute anything at all to taxonomy extraction. The price to pay is that for all these methods, including ours, the extracted relations are of somewhat mixed quality. While a significant fraction of the extracted relations are of the kind hyponym/hypernym (banana - fruit), many relations reflect that one term is somehow an aspect of the other term, e.g., disease - outbreak. In many applications these latter relations are of limited use.

The bulk of the existing very large body of work on taxonomy extraction therefore makes use of some external knowledge base or the other. Given the abundance of material, and the scope of this paper, we give only a brief overview here. For a more comprehensive treatment, see the recent survey by Uren *et al* [27] or the recent book by Buitelaar *et al* [6].

One of the simplest approaches, yet a very effective one, is the use of so-called *Hearst-patterns*, that is, patterns in the text that are likely to point to a hyponym/hypernym relationship [15] [16]. Examples are "such as", as in "fruits such as banana", or "and other", as in "banana and other fruits". Following this idea, numerous schemes with a more sophisticated linguistic parsing have been presented. For example, Woods [29] used a large morphological knowledge base to extract asymmetric term relations from compound phrases, such as "oak tree" or "tree root". In a similar vein, Anick and Tipirneni [2] measured the *dispersion* of terms, that is, the number of compound phrases a particular term appears in; terms with a large dispersion would be in the upper layer of their two-level hierarchy. Maedche and Staab [21] combine established tools for shallow text processing and association rule mining.

A simple and elegant way to boost approaches based on linguistic patterns is to search for the patterns not on the collection, for which a taxonomy is

to be constructed, but on the Web (or a similarly large and diverse external collection). The basic idea is then to formulate a number of hypothetical relations as keyword (phrase) queries, for example, `"banana is a fruit"` or `"banana is an animal"`, and then assess their validity by the number of hits obtained. Two recent systems built on this idea are Pankow [5] and KnowItAll [13].

More sophisticated systems enhance the set of relations obtained by one or more of these techniques, and/or relations input by a human, by various kinds of *bootstrapping* techniques. The Snowball system, for example, tries to learn new extraction patterns from the relations it already knows, then applies these patterns to obtain new relations, from these tries to learn more patterns, and so on [1].

Finally, there are many systems whose primary goal is not the extraction of a term taxonomy but to assist the user in identifying meaningful relations by offering few promising options. Examples are OntoMat [28] and SemTag [7].

2 Our Algorithm for Computing a Term Taxonomy

In this section, we give a mathematical criterion for determining the relation of each term pair. The criterion is based on a sequence of low-rank approximations of the term-term similarity matrix.

2.1 The Term-Term Similarity Matrix

Our approach requires a matrix that specifies for each term pair a similarity. A simple way to obtain similarities between each pair of terms is to multiply the document-term matrix $\mathbf{A}$ by its transpose, that is, compute $\mathbf{S} = \mathbf{A}^T\mathbf{A}$, and take the entry $\mathbf{S}_{ij}$ as a measure for the similarity between term i and term j. If we normalize the columns (terms) of $\mathbf{A}$ prior to computing $\mathbf{S}$, then all diagonal entries (standing for the similarities of terms with themselves) will be exactly 1. Two terms which never co-occur in any document, have a similarity of zero.

Other common measures for term-term similarity are as follows. For the so-called *Pearson correlation*, the mean row (document) is subtracted from every row (document) of $\mathbf{A}$, and columns (terms) are normalized by dividing by their standard deviation from the mean norm (after subtracting the mean document from every document, the columns have mean zero, so this is equivalent to dividing the columns by their norms) of a column (term). Nanas et al. [22] count the number of term co-occurrence in sliding windows of fixed length, giving more weight to pairs of terms appearing close each other. Park et al. [24] use a Bayesian network.

We remark that computing term-term similarities from term co-occurrence information makes sense when each document of the given collection is on a single topic: Two terms that co-occur frequently with each other must then refer to a common topic and are thus related. If the documents are not believed to be single-topic, we can always break them up into smaller single-topic chunks, and call these our documents.

The method we present here does not rely on a particular measure of similarity or distance. The only requirement is an estimate of the similarity between any two index terms, represented by a symmetric matrix $\mathbf{S}$.

To determine the kind of relations of term pairs, we will look at all the *low-rank approximations* of the term-similarity matrix $\mathbf{S}$. Let $\mathbf{S} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$ be the eigenvector (Schur) decomposition of $\mathbf{S}$. Then $\mathbf{S_k} = \mathbf{V}(k)\mathbf{\Sigma}(k)\mathbf{V}(k)^T$ is the best rank-k approximation of $\mathbf{S}$ in terms of both Frobenius and L_2-norm, where $\mathbf{S}(k)$ is the diagonal matrix holding the k largest eigenvalues, and $\mathbf{V}(k)$ is the matrix consisting of those k columns from $\mathbf{V}$ pertaining to these eigenvalues.

2.2 Term Validity Rank

We define the similarity $\mathrm{sim}_k(i, j)$ between the i-th and j-th terms at dimension k by the entry $\mathbf{S}(k)_{ij}$. To investigate how the similarity of a term changes with the dimension k, we define the notion of *similarity curve*, which is defined as *relatedness curve* in [3].

Definition 1 (Similarity Curve). *The* similarity curve *of two terms* $\mathtt{t}_i$ *and* $\mathtt{t}_j$ *is defined as the plot of the function*

$$k \mapsto \mathrm{sim}_k(i, j) = \mathbf{S}(k)_{ij}$$

We seek a representation that is sufficiently detailed to encompass enough information for a term to be correctly represented, without being so detailed as to distinguish between terms with essentially the same meaning. The following definition uses the notion of a term being more similar to itself than to any other term:

Definition 2 (Validity). *A term* $\mathtt{t}$ *is correctly represented in the* k*-order approximation of the similarity matrix only if it is more similar to itself than to any other term, that is,* $\mathrm{sim}_k(\mathtt{t}, \mathtt{t}) \geq \mathrm{sim}_k(\mathtt{t}, \mathtt{t'})$ *for any other term* $\mathtt{t'} \neq \mathtt{t}$*. The term* $\mathtt{t}$ *is then said to be* valid *at rank* k*.*

Note that for a term $\mathtt{t}$, for a rank $k < N$, $\mathrm{sim}_k(\mathtt{t},\mathtt{t})$ is usually less than $\mathrm{sim}_N(\mathtt{t},\mathtt{t})$, but still more than $\mathrm{sim}_k(\mathtt{t},\mathtt{t'})$ for any other term $\mathtt{t'}$ when $\mathtt{t}$ is valid.

It is useful to define the rank below which a term ceases to be valid:

Definition 3 (Validity Rank). *A term* $\mathtt{t}$ *is optimally represented in the* k*-order approximation of the similarity matrix if* $k - 1$ *is the largest value for which it is not valid. Note that it implies that the term is valid at rank* k*, which is the* validity rank *of term* $\mathtt{t}$ *and is denoted* $\mathrm{rank}(\mathtt{t})$*.*

In practice it might happen for some terms that validity is achieved and lost successively for a short range of ranks. It is not clear whether this is due to a lack of precision in the numerically sensitive eigenvalue decomposition process or to theoretical reasons.

The definition of validity was experimentally illustrated in [8] where all the documents containing a specific term `a` were replicated in the database with `a`

replaced by some new term a'. The query composed of the term a was shown to return in alternation a and a' versions of the documents as long as the rank k of the approximation was below the validity rank of a. Beyond the validity rank, version of the documents containing the term a were returned first, suggesting that the representation of that term was ambiguous below rank(a), and unambiguous beyond it. This shows that if the rank of the similarity approximation matrix and the validity rank of a term used as a single word query coincide, then retrieval precision[1] is optimal. This justifies Definition 2 a posteriori. An extension to more than one term queries showed mixed results in [9]. A theoretical justification of the experimental result obtained in [8] was presented in [3].

2.3 Term Taxonomy

In the experiment described above, we observed that terms a and a' were not correctly distinguished in the k-dimensional latent concept space if k is inferior to the validity rank of a[2]. This shows that 1) the two terms bear a common meaning to a certain extent, 2) the common meaning is more general than the meaning of any of the two terms. For these two reasons, we call the common meaning the *concept*[3] shared by the two terms.

Moreover, we know by Definition 3 that below their validity rank, a and a' are more similar to some other terms than to themselves. If they are both more similar to a common term c valid at rank k, the representation of this term better covers the concept common to a and a': We say that a and a' share the common concept $\mathtt{c}^*$ where the notation $\mathtt{c}^*$ is used to recall the difference between the representation of the single term document at full rank and at its validity rank.

Definition 4 (Concept of a Term). *A concept* $\mathtt{c}^*$ *associated to term* c *is a* concept of term $\mathtt{a}$ *if* rank(c) < rank($\mathtt{a}$) *and if for some rank* k *such that* rank(c) $\leq k <$ rank($\mathtt{a}$)*,* $\mathtt{a}^*$ *is more similar to* $\mathtt{c}^*$ *than to itself.*

The requirement that rank(c) < rank(a) ensures that $\mathtt{a}^*$ is never a concept of $\mathtt{c}^*$ if $\mathtt{c}^*$ is a concept of $\mathtt{a}^*$. If we associate terms to nodes and add directed links from the terms to their concepts, we obtain a directed acyclic graph (DAG). In practice, there is a whole range of ranks between rank(c) and rank(a) where concept $\mathtt{a}^*$ points to its concept $\mathtt{c}^*$, and we keep only the largest one to construct the graph. By identifying the concepts associated to all the terms, we can construct a taxonomy. This is illustrated in Section 4.

There is a typically a range of ranks between rank(c) and rank(a) where concept $\mathtt{a}^*$ points to its concept $\mathtt{c}^*$. This motivates the following definition:

[1] We refer to the traditional definition of precision and recall.

[2] Terms a and a' being perfectly related, they have the same validity rank, as we will show in Section 3.

[3] This concept differs from the notion of *latent concept* popularized by Latent Semantic Analysis.

Definition 5 (Coverage of a Link). *Define* $k_{\min}$ *and* $k_{\max}$ *as the minimum and maximum* k *for which* $\mathtt{c}^*$ *is a concept of term* $\mathtt{a}$*. Since they verify* $\text{rank}(\mathtt{c}) \leq k_{\min} \leq k_{\max} < \text{rank}(\mathtt{a})$*, we can define the normalized* coverage *of the link between the two concepts as the ratio*

$$coverage = \frac{k_{\max} - k_{\min} + 1}{\text{rank}(\mathtt{a}) - \text{rank}(\mathtt{c})}$$

The coverage has values in $]0, 1]$.

The coverage reflects "how long", with respect to the possible range defined by rank(c) and rank(a), the valid term was a concept for the other term. We will see when we illustrate the hierarchy building procedure in Section 4 that the coverage is a good predictor of interesting links.

3 Theoretical Underpinning of our Algorithm

We next justify the notion of validity rank via a simple model. Intuitively, if a concept $\mathtt{c}^*$ associated to a term c is a concept for a set of terms T then we expect c to occur in the documents in which any of the terms $\mathtt{a} \in \mathtt{T}$ is present. We define a notion of a concept $\mathtt{c}^*$ being a *perfect concept* for two other terms, which have symmetrical co-occurrence patterns and are called perfectly related terms in [3].

Definition 6 (Perfect Concept). *Let* $\mathbf{A}$ *be a* $D \times N$ *document-term matrix with* D *documents and* N *terms. Without loss of generality, let us assume that the terms* c*,* a *and* a' *correspond to the last three columns of* $\mathbf{A}$*. Then,* a *and* a' *are said to be* perfectly related *to each other and the concept* $\mathtt{c}^*$ *associated to* c *is said to be a* perfect concept *for* a *and* a' *if, for some permutation of the rows,*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_1} & \mathbf{a_1} & \mathbf{a_1} & \mathbf{0} \\ \mathbf{A_1} & \mathbf{a_1} & \mathbf{0} & \mathbf{a_1} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where $\mathbf{A_1}$ *is a sub-matrix of dimension* $d \times (N-3)$*,* $\mathbf{a_1}$ *is column vector of size* d *and* $\mathbf{B}$ *is a sub-matrix of dimension* $(D - 2d) \times (N - 3)$*, for some* d *with* $0 \leq d < D/2$.

The following two lemmas say that a perfect concept $\mathtt{c}^*$ of a pair of perfectly related terms a and a' induce a particular substructure in the eigenvectors, which in turn implies that c is always more similar to itself than to a or a' while for a large range of dimension k a and a' are more similar to c than to themselves. Hence $\mathtt{c}^*$ is a concept for a and a'.

Lemma 1. *Let* $\mathbf{S}$ *be an* $N \times N$ *symmetric matrix such that*

$$\mathbf{S} = \begin{bmatrix} \mathbf{C} & \frac{2\beta}{\alpha}\mathbf{c}^T & \mathbf{c}^T & \mathbf{c}^T \\ \frac{2\beta}{\alpha}\mathbf{c} & \frac{2\beta^2}{\alpha} & \beta & \beta \\ \mathbf{c} & \beta & \alpha & 0 \\ \mathbf{c} & \beta & 0 & \alpha \end{bmatrix}$$

where $\mathbf{C}$ *is a symmetric sub-matrix of dimension* $(N-3)\times(N-3)$, $\mathbf{c}$ *is a row vector of size* N, α *and* β *are scalers. Then,*

1. *The vector* $\mathbf{v} = (0,\ldots,0,0,-\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$ *is an eigenvector of* $\mathbf{S}$ *with eigenvalue* α.
2. *The vector* $\mathbf{v'} = (0,\ldots,0,-\frac{\alpha}{\beta},1,1)$ *is an unnormalized eigenvector of* $\mathbf{S}$ *with eigenvalue 0.*
3. *All other eigenvectors* $\mathbf{u}$ *of* $\mathbf{S}$ *are of the form* $\mathbf{u} = (u_1,\ldots,u_{N-3},2\frac{\beta}{\alpha}x,x,x)$ *for some* x.

Proof. The proofs of parts 1 and 2 are straightforward, because $\mathbf{Sv} = \alpha\mathbf{v}$ and $\mathbf{Sv'} = 0$. If $\mathbf{u}$ is any other eigenvector of $\mathbf{S}$ and if the last three entries of $\mathbf{u}$ are u_{N-2}, u_{N-1} and u_N, then $u_{N-1} = u_N$ because $\mathbf{u}$ is orthogonal to $\mathbf{v}$. Also, since $\mathbf{u}$ is orthogonal to $\mathbf{v'}$, we have $\frac{\alpha}{\beta}u_{N-2} = u_{N-1} + u_N$, hence part 3 of lemma 1 follows.

Lemma 2. *For a document-term matrix* $\mathbf{A}$ *as in Definition 6, the correlation matrix* $\mathbf{S} = \mathbf{A}^T\mathbf{A}$ *has the form as described in lemma 1. This property is invariant of whether the columns (terms) of* $\mathbf{A}$ *are normalized before computing* $\mathbf{S}$ *or not.*

Proof. The correlation matrix $\mathbf{S} = \mathbf{A}^T\mathbf{A}$ has the form as in 1 with $\mathbf{C} = 2\mathbf{A_1}^T\mathbf{A_1} + \mathbf{B}^T\mathbf{B}$, $\mathbf{c} = \mathbf{A_1}^T\mathbf{a_1}$, and $\alpha = \beta = \mathbf{a_1}^T\mathbf{a_1}$. If the columns of $\mathbf{A}$ are normalized first, then the normalized matrix becomes

$$\mathbf{A'} = \begin{bmatrix} \mathbf{A'_1} & \frac{\mathbf{a_1}}{\sqrt{2}|\mathbf{a_1}|} & \frac{\mathbf{a_1}}{|\mathbf{a_1}|} & \mathbf{0} \\ \mathbf{A'_1} & \frac{\mathbf{a_1}}{\sqrt{2}|\mathbf{a_1}|} & \mathbf{0} & \frac{\mathbf{a_1}}{|\mathbf{a_1}|} \\ \mathbf{B'} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

for some sub-matrices $\mathbf{A'_1}$ and $\mathbf{B'}$. Then, $\mathbf{S} = \mathbf{A'}^T\mathbf{A'}$ is of the form as in Lemma 1 with $\mathbf{C} = 2\mathbf{A'_1}^T\mathbf{A'_1} + \mathbf{B'}^T\mathbf{B'}$, $\mathbf{c} = \frac{1}{\sqrt{2}}\mathbf{A'_1}^T\mathbf{a_1}$, $\alpha = 1$ and $\beta = \frac{1}{\sqrt{2}}$, hence the lemma.

Suppose $\mathbf{A}$ is a document-term matrix as in Definition 6, terms `a` and `a'` are perfectly related to each other and `c`* is a perfect concept for `a` and `a'`. Using the similarity curves of the terms `c` and `a` we show that `c`* is a concept for `a` as defined in Definition 4. If the correlation matrix $\mathbf{S} = \mathbf{A}^T\mathbf{A}$ is computed after normalizing the columns of $\mathbf{A}$, by Lemma 2, $\mathbf{S}$ has the form as shown in Lemma 1 with $\alpha = 1$ and $\beta = \frac{1}{\sqrt{2}}$. From Lemma 1, we also know that $\mathbf{v} = (0,\cdots,0,-\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$ is an eigenvector of $\mathbf{S}$ with eigenvalue $\alpha = 1$. Also, $\mathbf{v'} = (0,\cdots,0,-\frac{1}{\sqrt{2}},\frac{1}{2},\frac{1}{2})$ is the normalized form of another eigenvector of $\mathbf{S}$ with eigenvalue 0. Let 1 be the k-th eigenvalue of $\mathbf{S}$. From Lemma 1 the three rows of $\mathbf{V}$ corresponding to terms `c`, `a` and `a'` are of the form

$$\begin{array}{lccccccc} \texttt{c}: & \sqrt{2}x_1 \ldots & \sqrt{2}x_{k-1} & 0 & \sqrt{2}x_{k+1} \ldots & \sqrt{2}x_{N-1} & -\frac{1}{\sqrt{2}} \\ \texttt{a}: & x_1 \ldots & x_{k-1} & -\frac{1}{\sqrt{2}} & x_{k+1} \ldots & x_{N-1} & \frac{1}{2} \\ \texttt{a'}: & x_1 \ldots & x_{k-1} & \frac{1}{\sqrt{2}} & x_{k+1} \ldots & x_{N-1} & \frac{1}{2} \end{array}$$

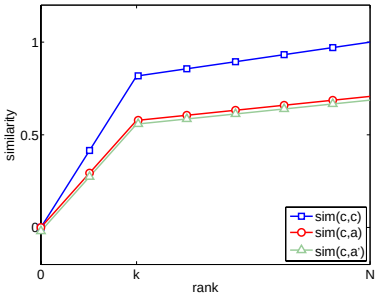


Fig. 1. Similarity curves of `c` with `a`, `a'` and itself. The curves for (`c`,`a`) and (`c`,`a'`) actually overlap, but for clarity we have shown them separately.

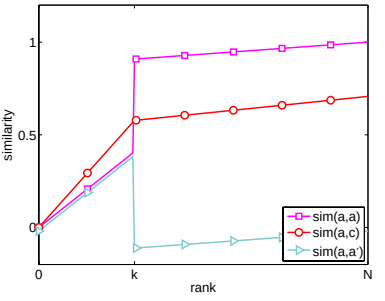


Fig. 2. Similarity curves of `a` with `c`, `a'` and itself. Until rank k, the curves of (`a`,`a`) and (`a`,`a'`) overlap, but for clarity we have drawn the lines separately.

From this particular substructure in the eigenvectors, we obtain similarity curves of `c` with `a`, `a'` and itself as in Figure 1, and similarity curves of `a` with itself and two other terms as in Figure 2. To draw these similarity curves, we do not take the eigenvalues of $\mathbf{S}$ into account because, for a general matrix like $\mathbf{S}$, all the eigenvalues cannot be determined, and as discussed in [3], the use of eigenvalues does not change the overall behavior of the similarity curves. Also, to illustrate the relative behavior of the curves clearly, we draw straight lines between the main breakpoints of the curves.

Let us assume that the terms `c`, `a` and `a'` are not related to any other term, so that their validity ranks depend only on their mutual similarities. Since the similarity curves of (`c`,`c`) is above the curves for (`c`,`a`) and (`c`,`a'`) for all ranks, the term `c` is valid at all ranks greater than or equal to 1 and hence the validity rank of `c` is 1. On the other hand, we observe from 2 that until dimension k, sim(`c`, `a`) is larger than sim(`a`, `a`), so term `a` is not valid until rank k. However, for ranks k or greater, the similarity curve of (`a`,`a`) rises above the curves of (`a`,`c`) and (`a`,`a'`) (Figure 2) and so `a` is valid for all ranks $k' \geq k$. Hence the validity rank of `a` is k. Here, `c`* is a concept for `a`* for all ranks k' for $0 \leq k' < k$.

4 Numerical Experiments

There are no standard procedures to evaluate hierarchies although some attempts have been made [18]. Beyond the fact that evaluation is difficult even when a group of volunteers is willing to participate, it also depends on the task the hierarchy is designed for. For example, the measure used in [18] could not be applied here as the scoring is based on an estimate of the time it takes to find all relevant documents by calculating the total number of menus –this would be term nodes in this work– that must be traversed and the number of documents that must be read, which bears no analogy to this work.

As mentioned in the Introduction, we expect PCA to uncover symmetric and asymmetric relations between terms. We can divide further asymmetric relations

in two types: The first one is semantic and can be found in dictionaries like WordNet[4]. These are relations that derives from the definition of the terms like "cat" and "animal" for example. The other kind of relation we expect to uncover is more circumstantial but equally interesting like, for example, "Rio de Janeiro" and "Carnival". These two words share no semantic relation, but associating them makes sense. To evaluate the PCA hierarchy, we chose to compare the links it extracts from the document collection associated with the Open Directory Project[5] to the original, edited hierarchy. To identify the ability of PCA to extract "semantic" relations, we compare it with WordNet.

The *Open Directory Project* (ODP) is the most comprehensive human edited directory of the Web. We extracted the hierarchies below the entries *Shopping* and *Science*. Out of the 104,276 and 118,584 documents referred by these categories, we managed to download 185,083 documents to form the database we use.

Documents were processed with a language independent part-of-speech tagger[6] and terms replaced by their lemmata. We extracted only adjectives and substantives to form the bag-of-word representations. Low and high frequency terms as well as stopwords were discarded unless they appeared in the ODP hierarchy. Documents were divided in blocks of 25 terms to reduce the confusion of topics inside a same document (Section 2).

A path in the ODP hierarchy is composed as a series of topics, from the most generic to the most specific. An example of such a path is "Health/Beauty/-Bath_and_Body/Soap". We discard concepts described as a sequence of terms. For example, the previous sequence is transformed into "Health/Beauty/Soap". The hierarchy is then decomposed into direct links – i.e. relations that exist between adjacent terms – and indirect links where relations between terms belong to the same path. The direct links in our example are Health ← Beauty and Beauty ← Soap and the set of transitive links is composed of the former links more Health ← Beauty.

In order to test the stability of the discovered links, we *bootstrapped* [12] the document database. The method consists in picking randomly with replacement 185,083 documents from the original database to form a new correlation matrix before deducing a new set of links. This process is repeated ten times. The number of replications where a particular link appears reflects its stability with respect to variations in the database. We say that a link is *stable* when the relationship between the two terms held the ten times, and in the opposite case it is said to be *unstable*. For the science and shopping topics, half of the links are stable.

This stability analysis permits us to discover terms that share an asymmetric relation which direction changes from one re-sampling to the other (unstable links). This points out terms with a symmetric relation like alluded in the Introduction. Table 1 shows an unedited list of such relations as extracted from the *Shopping* ODP database.

[4] `http://wordnet.princeton.edu`
[5] `http://www.dmoz.org`
[6] TreeTagger: `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger`

Table 1. Symmetric relations in the *Shopping* ODP database

woodwind	clarinet	aikido	judo	beef	meat	judo	karate
anarchism	libertarianism	mazda	nissan	new	used	feb	jan
crab	shrimp	effect	result	woodwind	bassoon	bassoon	woodwind
thursday	wednesday	lobster	shrimp	july	june	aikido	jujitsu
impressionism	surrealism	june	september	mg	new	florist	flowers
classification	confirmed	nov	oct	billiard	dart	dec	oct
expressionism	surrealism	judo	jujitsu	necktie	scarves	nissan	toyota
crab	lobster	flour	grain	august	june	grain	flour
clarinet	woodwind	ingredient	soap	english	word	skiing	snowboard

By analogy with the Information Retrieval measures, we define *recall* as the proportion of links in the original hierarchy that the PCA method manages to retrieve automatically. The *precision* is defined as the proportion of ODP links present in the set of PCA links. If we denote by H the set of links in the ODP human edited hierarchy and by A the set in the PCA automatic hierarchy, recall and precision become

$$recall = \frac{|H \cap A|}{|H|} \qquad precision = \frac{|H \cap A|}{|A|}$$

Recall answers the question "How many ODP link do I retrieve automatically?", while precision answers "What is the concentration of ODP links among all the PCA links?"

To fix ideas, we intended to compare our method with the most popular one, from Sanderson and Croft [25], but the results were so poor that we abandoned the idea. Their method use the co-occurrence information to identify a term that subsumes other terms. More specifically, a term u is said to subsume another term v if the documents in which v occurs belong to a subset of the documents in which u occurs. Given that a more frequent term tends to be more general [26], *subsumption hierarchies* organize terms in a 'general to specific' manner. For two terms x and y, x subsumes y whenever $P(x|y) \geq \Theta$ and $P(y|x) < P(x|y)$ where $P(x|y)$ is the probability that term x occurs in a document where y occurs. We first tried to use the same division of documents in sequences of 25 terms that we applied on ODP documents to avoid topic heterogeneity, but such a small window of terms proved detrimental for the algorithm and we finally decided to use the full documents. Sanderson and Croft set the parameter Θ to 0.8 in the original paper, but we varied it to study its impact on precision and recall. The results were quite poor. Over all possible settings with respect to the stability, the minimum Θ and the hierarchies we compared with, the recall we achieved topped around 5% for a precision of approximately 1%. The best precision amount to 12.5% but the corresponding recall is only 0.03%!

In the remaining of this section, we compute the proportion of direct and indirect links present in ODP that we retrieve automatically with our Principal Component Analysis method. We also study the impact of link stability and coverage (Definition 5). In the last part, the same set of discovered links is compared with the WordNet database. Note that a large intersection between human and automatically generated links increases the confidence on the validity

Table 2. The first 30 direct links in *Shopping* and *Science* databases, ordered by decreasing coverage and limited to the stable links. Links in bold are in the ODP database.

Shopping			Science		
alberta	→	**canada**	**humidor**	→	**cigar**
monorail	→	lighting	**cuban**	→	**cigar**
criminology	→	sociology	**alberta**	→	**canada**
prehistory	→	**archaeology**	**cuckoo**	→	**clock**
romanian	→	slovenian	**grandfather**	→	**clock**
gravitation	→	relativity	fudge	→	chocolate
forensics	→	forensic	**soy**	→	**candle**
aztec	→	maya	**putter**	→	**golf**
karelian	→	finnish	**quebec**	→	**canada**
oceania	→	asia	racquetball	→	racket
transpersonal	→	**psychology**	**tasmania**	→	**australia**
etruscan	→	greek	airbed	→	mattress
barley	→	wheat	**glycerin**	→	**soap**
papuan	→	eastern	**snooker**	→	**billiard**
quebec	→	**canada**	**housebreaking**	→	**dog**
cryobiology	→	cryonics	waterbed	→	mattress
soho	→	solar	oceania	→	asia
catalysis	→	**chemistry**	tincture	→	herbal
geotechnical	→	engineering	**gunsmithing**	→	**gun**
iguana	→	**lizard**	chrysler	→	chevrolet
sociologist	→	**sociology**	equestrian	→	horse
olmec	→	maya	flamenco	→	guitar
oceanographer	→	**oceanography**	**pistachio**	→	**nut**
canine	→	dog	condiment	→	sauce
neptunium	→	plutonium	appraiser	→	estate
lapidary	→	mineral	**salsa**	→	**sauce**
raptor	→	**bird**	**ontario**	→	**canada**
ogham	→	irish	volkswagen	→	volvo
governmental	→	**organization**	arthropod	→	insect
forestry	→	forest	bulldog	→	terrier

of the automatic method, but it does not invalidate the automatic links absent from edited hierarchy because documents and topics can be organized in a variety of equally good ways. This is corroborated in Table 2 where links absent from ODP are in normal font.

4.1 Coverage and Stability of Direct Links

Coverage is perceived as a relevant indicator of link quality because it reflects the strength that unite the two terms linked by a hierarchical relation. In Table 3, the number of links discovered from the *Science* documents are reported as a function of the minimum coverage in both the stable and unstable cases. We see that 70% and 80% of the links have a coverage lower than 20%. Discarding all the links below this level of coverage results in the loss of only 30% and 33% of ODP links.

The stability is also an important selection criterion. We observe that if we consider all the PCA links, stable or not, we retrieve 551 of the original 2,151 ODP links present in Science. If we select only the stable links, we retrieve 436 ODP links, but the total number of PCA links is divided by two from 28,859 to 14,266. Some of the links present in the ODP hierarchy are lost, but more than half of the PCA links are discarded. A similar conclusion holds when varying the

Table 3. Number of links discovered by PCA in *Science* documents as a function of the coverage and, in parenthesis, the size of the intersection with the 2,151 *Science* ODP links

	stable	unstable
0%	14,266 (436)	28,859 (551)
20%	3,832 (308)	5,850 (368)
40%	1,867 (251)	2,831 (261)
60%	1,095 (166)	1,676 (198)
80%	644 (115)	998 (138)
99%	218 (59)	294 (65)

coverage minimum threshold. This justifies stability as an important criterion for selecting a link.

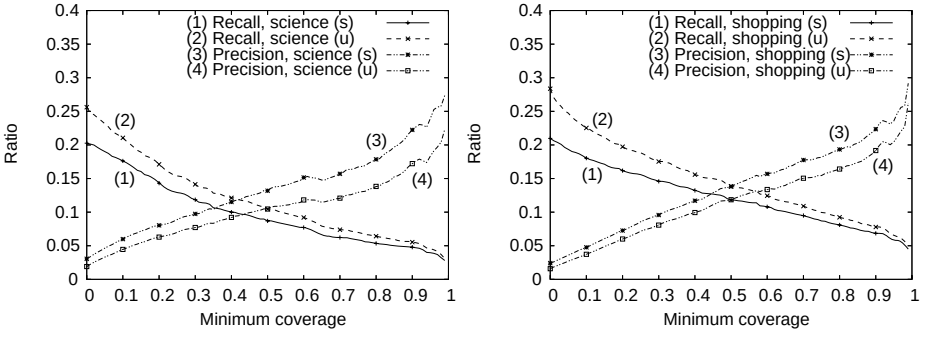


Fig. 3. Comparison of the PCA stable (s) and unstable (u) links with the ODP hierarchy on the Science and Shopping topics. *Recall* is the proportion of links in the original ODP hierarchy rediscovered by PCA. *Precision* is the proportion of ODP links among those retrieved by PCA. The x-axis is the coverage ratio: For a given value c, the PCA links we consider are those whose coverage is superior to c.

Fig. 3 offers a global view of the impacts of stability and coverage on recall and precision for topics *Science* on the left and *Shopping* on the right. The portion of common links is significantly larger when the coverage is closer to its maximum. On both graphs, if we select only links with a coverage superior to 0.8, one tenth of the links in A are present in ODP. When varying the coverage threshold from 0 to 1, precision increases and recall decreases almost always. This means that coverage is a good predictor of the link "relevance". This was verified empirically as well by inspecting some part of the discovered links ordered by coverage.

Summarizing, stability and coverage are both important predictors of link quality.

4.2 Transitive Links

Some links present in ODP might appear as combination of links in PCA and vice-versa. We already explained how ODP was processed to obtain these links.

For PCA, we create a link between two terms if there is a path from one term to another. A link is said to be direct if it appears in the original hierarchy, and indirect if it was discovered by transitivity. A set of links is transitive if it includes both direct and indirect links.

The coverage being a good indicator of the link quality, we try to extend this notion to transitive links. We found experimentally that the minimum coverage of all the traversed links led to the best results: An indirect link is penalized if all the paths between the two terms traverse a link with a low coverage.

A study of the effect of coverage and stability on precision and recall is reported in Fig. 4 (left) where we aggregated the results over the science and shopping topics, and compared the direct and transitive ODP and PCA links. The results being similar for both topics, there is no need to treat them separately. Precision and recall when both links set are either transitive or direct (PCA, ODP and PCA+, ODP+ curves on Fig. 4, left) are very similar: This shows that precision is not much affected by the new PCA indirect links (around 38% more links, from 31,611 to 43,632) while recall is not much affected by the new ODP links (around 126% more links, from 4,153 to 9391). It is interesting also to observe that among the 2,297 links common to the transitive PCA and ODP sets, 1,998 are present in the direct PCA set. This is reflected on Fig. 4 (ODP+, PCA plot) where the corresponding precision curve is significantly superior while recall is less affected. This suggests that the indirect links of PCA did not contribute much.

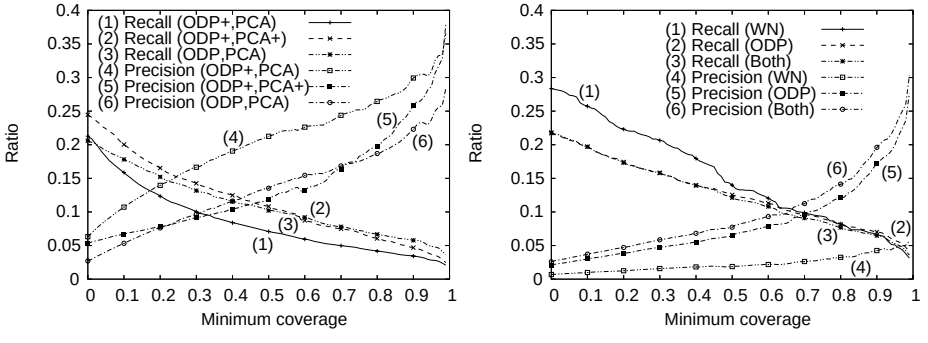


Fig. 4. Left: Comparison of the PCA direct links (PCA) and transitive links (PCA+) with the ODP direct links (ODP) and transitive links (ODP+). Right: Comparison with different hierarchies. Only stable and transitive links are considered.

In conclusion, the new definition of coverage as the minimum on the path of traversed links proves a good selection indicator, as the precision increases with the coverage threshold. The manually derived and the PCA hierarchies share a significant amount of links and it seems that PCA is successful in discovering relations between terms. This is a specially good result given that the ODP directory is only one among numerous possible ways of organizing the documents in the database.

4.3 Comparison with WordNet

WordNet is a lexical database for the English language while the PCA hierarchy method necessarily reflects the document collection it is extracted from. It is nevertheless interesting to compare these hierarchies and investigate to what point PCA is able to detect lexical relations.

To obtain the links from WordNet, we followed three different relationships: the hypernyms (generalization), the holonyms (an object is a part of) and the synonyms. For each term, we first computed the sets of its synonyms. We computed the transitive closure of this set with respect to each relationship separately. We then selected all the links from the original term to one of the terms in the three computed sets. We restricted the set of links to those only containing terms belonging to the ODP topic: This created two sets of links, one for science and one for shopping.

There are few links common to ODP and WordNet. For the *Science* topic, only 464 links were common, representing 10% and 24% of the links present in WordNet and ODP respectively. For shopping, we found 354 common links representing 8% and 11% of WordNet and ODP respectively.

In Fig. 4 (right), we compare the PCA hierarchy with ODP, WordNet and with the union of WordNet and ODP. We use the transitive links for the three of them. We observe that the PCA hierarchy is closer to ODP than to WordNet in terms of precision: the precision for WordNet varies between 1% and 5% while it varies between 2% and 28% for ODP. This reflects the fact that PCA and ODP are based on the same set of documents. When ODP and WordNet hierarchies are merged, we observe a small overall increase of precision, lower than the simple aggregation of the ODP and WordNet precision curves because some links are common.

Recall values are similar for the three hierarchies. This is a good result since, as stated above, the intersection between WordNet and ODP is relatively small.

5 Conclusions

We have shown a way to use eigenvector decomposition for the automatic extraction of a term taxonomy. We have shown some mathematical soundness properties of our approach, and our experiment gave promising results.

The mathematic model could be extended by analysing the effect of perturbation on the special matrix considered in Section 3. We would expect matrix perturbation theory, as pioneered in [23] and used in a number of subsequent works, to be helpful for this task.

Our primary goal in this work was the automatic extraction of a term taxonomy. It would be interesting to use this taxonomy for information retrieval tasks such as ad-hoc retrieval. We would expect asymmetric relations to give potentially better results than symmetric ones.

Finally, it would be interesting to investigate the possibility of combining the fully automatic eigenvector approach, explored in this paper, with some kind of external knowledge base.

References

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *5th Conference on Digital Libraries (DL'00)*, 2000.

[2] P. G. Anick and S. Tipirneni. The paraphrase search assistant: terminological feedback for iterative information seeking. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 153–159, New York, NY, USA, 1999. ACM Press.

[3] H. Bast and D. Majumdar. Why spectral retrieval works. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18. ACM, 2005.

[4] S.-L. Chuang and L.-F. Chien. A practical web-based approach to generating topic hierarchy for text segments. In *CIKM '04: Proceedings of the Thirteenth ACM conference on Information and knowledge management*, pages 127–136, New York, NY, USA, 2004. ACM Press.

[5] P. Cimiano, G. Ladwig, and S. Staab. Gimme' the context: context-driven automatic semantic annotation with c-pankow. In *14th International Conference on the World Wide Web (WWW'05)*, pages 332–341, 2005.

[6] P. B. P. Cimiano and B. Magnini. *Ontology Learning from Text: Methods, Evaluation and Applications (Volume 123: Frontiers in Artificial Intelligence and Applications)*. IOS Press, 2005.

[7] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, K. McCurley, S. Rajagopalan, A. Tomkins, J. Tomlin, and J. Zienberer. A case for automated large scale semantic annotation. *J. Web Semantics*, 1(1), 2003.

[8] G. Dupret. Latent concepts and the number orthogonal factors in latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 221–226. ACM Press, 2003.

[9] G. Dupret. Latent semantic indexing with a variable number of orthogonal factors. In *Proceedings of the RIAO 2004, Coupling approaches, coupling media and coupling languages for information retrieval*, pages 673–685. Centre de Hautes Etudes Internationales d'informatique documentaire, C.I.D., April 26-28 2004.

[10] G. Dupret and B. Piwowarski. Deducing a Term Taxonomy from Term Similarities. In *ECML/PKDD 2005 Workshop on Knowledge Discovery and Ontologies*, 2005.

[11] G. Dupret and B. Piwowarski. Principal components for automatic term hierarchy building. In *Proceedings of the 13th International Symposium on String Processing and Information Retrieval (SPIRE 2006)*, LNCS 4209, pages 37–48. Springer, 2006.

[12] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, May, 15 1994.

[13] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[14] E. Glover, D. M. Pennock, S. Lawrence, and R. Krovetz. Inferring hierarchical descriptions. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 507–514, New York, NY, USA, 2002. ACM Press.

[15] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[16] M. A. Hearst. Automated discovery of wordnet relations. In e. Fellbaum, Christiane, editor, *WordNet: An Electronic Lexical Database, MIT Press*, May 1998.

[17] H. Joho, C. Coverson, M. Sanderson, and M. Beaulieu. Hierarchical presentation of expansion terms. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 645–649, New York, NY, USA, 2002. ACM Press.

[18] D. Lawrie and W. Croft. Discovering and comparing topic hierarchies. In *Proceedings of RIAO 2000*, 2000.

[19] D. Lawrie, W. B. Croft, and A. Rosenberg. Finding topic words for hierarchical summarization. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357, New York, NY, USA, 2001. ACM Press.

[20] D. J. Lawrie and W. B. Croft. Generating hierarchical summaries for web searches. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 457–458, New York, NY, USA, 2003. ACM Press.

[21] A. Maedche and S. Staab. Discovering conceptual relations from text. In *14th European Conference on Artifial Intelligence (ECAI'00)*, pages 321–325, 2000.

[22] N. Nanas, V. Uren, and A. D. Roeck. Building and applying a concept hierarchy representation of a user profile. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 198–204, New York, NY, USA, 2003. ACM Press.

[23] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: a probabilistic analysis. In *Proceedings PODS'98*, pages 159–168, 1998.

[24] Y. C. Park, Y. S. Han, and K.-S. Choi. Automatic thesaurus construction using bayesian networks. In *CIKM '95: Proceedings of the fourth international conference on Information and knowledge management*, pages 212–217, New York, NY, USA, 1995. ACM Press.

[25] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213, New York, NY, USA, 1999. ACM Press.

[26] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972. (Reprinted in Griffith, B. C. (Ed.) Key Papers in Information Science, 1980, and in Willett, P. (Ed.) Document Retrieval Systems, 1988).

[27] V. Uren, P. Cimiano, J. Iria, S. Handschuh, M. Vargas-Vera, E. Motta, and F. Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics*, 4(1):14–28, 2006.

[28] R. Volz, S. Handschuh, S. Staab, L. Stojanovic, and N. Stojanovic. Unveiling the hidden bride: deep annotation for mapping and migrating legacy data to the semantic web. *Journal of Web Semantics*, 1(2):187–206, 2004.

[29] W. A. Woods. Conceptual indexing: A better way to organize knowledge. Technical report, 1997. Sun Labs Technical Report: TR-97-61.

Semi-automatic Construction of Topic Ontologies

Blaž Fortuna, Dunja Mladenič, and Marko Grobelnik

Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
blaz.fortuna@ijs.si, dunja.mladenic@ijs.si, marko.grobelnik@ijs.si
http://kt.ijs.si/

Abstract. In this paper, we review two techniques for topic discovery in collections of text documents (Latent Semantic Indexing and K-Means clustering) and present how we integrated them into a system for semi-automatic topic ontology construction. The *OntoGen* system offers support to the user during the construction process by suggesting topics and analyzing them in real time. It suggests names for the topics in two alternative ways both based on extracting keywords from a set of documents inside the topic. The first set of descriptive keyword is extracted using document centroid vectors, while the second set of distinctive keyword is extracted from the SVM classification model dividing documents in the topic from the neighboring documents.

1 Introduction

When working with large corpora of documents it is hard to comprehend and process all the information contained in them. Standard text mining and information retrieval techniques usually rely on word matching and do not take into account the similarity of words and the structure of the documents within the corpus. We try to overcome that by automatically extracting the topics covered within the documents in the corpus and helping the user to organize them into a topic ontology.

A topic ontology is a set of topics connected with different types of relations. Each topic includes a set of related documents. Construction of such an ontology from a given corpus can be a very time consuming task for the user. In order to get a feeling on what the topics in the corpus are, what the relations between topics are and, at the end, to assign each document to some certain topics, the user has to go through all the documents. We try to overcome this by building a special tool which helps the user by suggesting the possible new topics and visualizing the topic ontology created so far – all in real time. This tool in combination with the corpus visualization tools [1] described in [8] aims at assisting the user in a fast semi-automatic construction of the topic ontology from a large document collection.

We chose two different approaches for discovering topics within the corpora. The first approach is a linear dimensionality reduction technique, known as

[1] http://kt.ijs.si/blazf/software.html

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 121–131, 2006.

Latent Semantic Indexing (LSI) [5]. This technique relies on the fact that words related to the same topic co-occur together more often than words related to the different topics. The result of LSI are fuzzy clusters of words each describing one topic. The second approach we used for extracting topics is the well known k-means clustering algorithm [12]. It partitions the corpus into k clusters so that two documents within the same cluster are more closely related than two documents from two different clusters. We used these two algorithms for automatic suggestion of topics during the construction of the topic ontology.

This paper is organized as follows. Section 2 gives a short overview of the related work on building otologies. Section 3 gives an introduction to the text mining techniques we used. Details about our system are presented in Section 4, evaluation and users' feedback are presented in Section 5 followed by the future work and conclusions in Sections 6 and 7.

2 Related Work on Building Otologies

Different approaches have been used for building ontologies, most of them using mainly manual methods. An approach to building ontologies was set up in the CYC project [6], where the main step involved manual extraction of common sense knowledge from different sources. There have been some definitions of methodology for building ontologies, again assuming manual approach. For instance, the methodology proposed in [19] involves the following stages: identifying the purpose of the ontology (purpose, intended application, range of users), building the ontology, evaluation and documentation. The building of the ontology is further divided in three steps. The first is *ontology capture*, where key concepts and relationships are identified, a precise textual definition of them is written, terms to be used to refer to the concepts and relations are identified, the involved actors agree on the definitions and terms. The second step involves *coding of the ontology* to represent the defined conceptualization in some formal languages (committing to some meta-ontology, choosing a representation language and coding). The third step involves possible *integration* with existing ontologies. An overview of the methodologies for building ontologies is provided in [7], where several methodologies, including the above described one, are presented and analyzed against the IEEE Standard for Developing Software Life Cycle Processes viewing ontologies as parts of some software product.

Recently, a number of workshops at Artificial Intelligence and Machine Learning conferences (ECAI, IJCAI, ECML/PKDD) have focused on the problem of learning ontologies. Most of the work presented there addresses one of the following: a problem of extending an existing ontology WordNet using Web documents [1], using clustering for semi-automatic construction of ontologies from parsed text corpora [2], [16], learning taxonomic, eg., "isa", [4], and non-taxonomic, eg., "hasPart" relations [15], extracting semantic relations from text based on collocations [11], extracting semantic graphs from text for learning summaries [14].

The contribution of this paper to the field is that it presents a novel approach to semi-automatic construction of a very specific type of ontology – topic

ontology. Text mining techniques (e.g. clustering) were already proven successful when used at this step (e.g. [2], [16]) and in this paper we present a very tight integration of them with a manual ontology construction tool. This allows our system to offer support to the user during the whole ontology construction process.

3 Text Mining Techniques

Text Mining is fairly broad in its research, addressing a large range of problems and developing different approaches. Here we present only a very small subset of the available methods, namely only those that we found the most suitable for the problem addressed in this paper.

3.1 Representation of Text Documents

In order to use the algorithms we will describe later we must first represent text documents as vectors. We use a standard *Bag-of-Words* (BOW) approach together with *TFIDF* weighting [17]. This representation is often referred to as *vector-space model*. The similarity between two documents is defined as the cosine of the angle between their vector representations – *cosine similarity*.

3.2 Latent Semantic Indexing

Language contains many redundant information, since many words share common or similar meaning. For computer this can be difficult to handle without some additional information – background knowledge. *Latent Semantic Indexing* (LSI), [5], is a technique for extracting this background knowledge from text documents. It uses a technique from linear algebra called Singular Value Decomposition (SVD) and bag-of-words representation of text documents for detecting words with similar meanings. This can also be viewed as extraction of hidden semantic concepts or topics from the text documents.

LSI is computed as follows. First *term-document matrix* A is constructed from a given set of text documents. This is a matrix with bag-of-words vectors of documents as columns. This matrix is decomposed using SVD so that $A = USV^T$ where matrices U and V are orthogonal and S is a diagonal matrix with ordered singular values on the diagonal. Columns of matrix U form an orthogonal basis of a subspace in bag-of-words space and vectors with higher singular values carry more information. Based on this we can view vectors that form the basis as concepts or topics. The space spanned by these vectors is called *Semantic Space*.

3.3 K-Means Clustering

Clustering is a technique for partitioning data so that each partition (or cluster) contains only points which are similar according to some predefined metric. In the case of text this can be seen as finding groups of similar documents, that is documents which share similar words.

K-Means [12] is an iterative algorithm which partitions the data into k clusters. It has already been successfully used on text documents [18] to cluster a large document corpus based on the document topic and incorporated in an approach for visualizing a large document collection [10]. You can see the algorithm roughly in Algorithm 1.

Algorithm 1: K-Means.
Input: A set of data points, a distance metric, the desired number of clusters k
Output: Clustering of the data points into k clusters
(1) Set k cluster centers by randomly picking k data points as cluster centers
(2) **repeat**
(3) Assign each point to the nearest cluster center
(4) Recompute the new cluster centers
(5) **until** the assignment of data points has not changed

3.4 Keywords Extraction

We used two methods for extracting keywords from a given set of documents: (1) keyword extraction using centroid vectors and (2) keyword extraction using Support Vector Machines. We used these two methods to generate description for a given topic based on the documents inside the topic.

The first method works by using the centroid vector of the topic (centroid is the sum of all the vectors of the document inside the topic). The main keywords are selected to be the words with the highest weights in the centroid vector.

The second method is based on the idea presented in [3] which uses Support Vector Machine (SVM) binary classifier [13]. Let A be the topic which we want to describe with keywords. We take all the documents from the topics that have A as a subtopic and mark these documents as negative. We take all the documents from the topic A and mark them as positive. If one document is assigned both negative and positive label we say it is positive. Then we learn a linear SVM classifiers on these documents and classify the centroid of the topic A. Keywords describing the concept A are the words, which's weights in SVM normal vector contribute most when deciding if the centroid is positive (it belongs to the topic).

The difference between these two approaches is that the second approach takes into account the context of the topic. Let's say that we have a topic named 'computers'. When deciding what the keywords for some subtopic A are, the first method would only look at what the most important words within the subtopic A are and words like 'computer' would most probably be found important. However, we already know that A is a subtopic of 'computers' and we are more interested in finding the keywords that separate it from the other documents within the 'computers' topic. The second method does that by taking the documents from all the super-topics of A as a context and learns the most crucial words using SVM.

4 Semi-automatic Construction of Topic Ontologies

We view semi-automatic topic ontology construction as a process where the user is taking all the decisions while the computer helps by giving suggestions for the topics, automatically assigning documents to the topics and suggesting names for the topics. The suggestions are applied only when the users decides to do so. The computer also helps by visualizing the topic ontology and the documents.

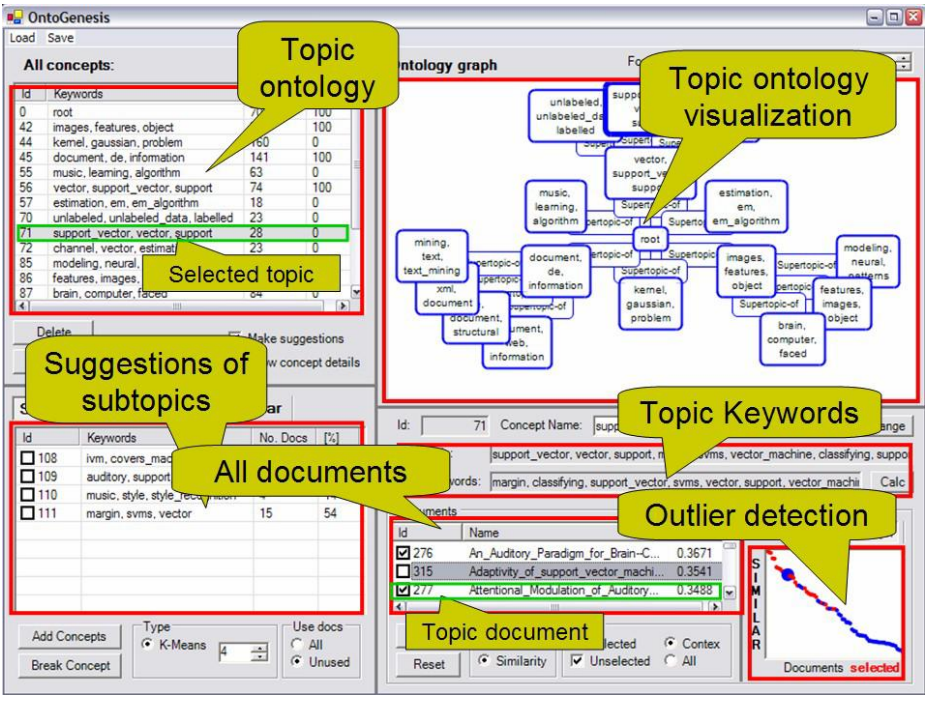


Fig. 1. Screen shot of the interactive system OntoGen for construction of topic ontologies

In Figure 1 you can see the main window of the interactive system *OntoGen* we developed. The system has three major parts that will be further discussed in following subsections. In the central part of the main window is a visualization of the current topic ontology (*Ontology visualization*). On the left side of the window is a list of all the topics from this ontology. Here the user can select the topic he wants to edit or further expand into subtopics. Further down is the list of suggested subtopics for the selected topic (*Topic suggestion*) and the list with all topics that are in relationship with the selected topic. At the bottom side of the window is the place where the user can fine-tune the selected topic (*Topic management*).

4.1 Ontology Visualization

While the user is constructing/changing topic ontology, the system visualizes it in real time as a graph with topics as nodes and relations between topics as edges. See Figures 1, 2 and 3 for examples of the visualization.

4.2 Topic Suggestion

When the user selects a topic, the system automatically suggests what the subtopics of the selected topic could be. This is done by LSI or k-means algorithms applied only to the documents from the selected topic. The number of suggested topics is specified by the user. Then, the user selects the subtopics he finds reasonable and the system automatically adds them to the ontology with relation 'subtopic-of' to the selected topic. The user can also decide to replace the selected topic with the suggested subtopics. In Figure 1 you can see how this feature is implemented in our system.

4.3 Topic Management

The user can manually edit each of the topics he added to the topic ontology. He can change which documents are assigned to this topic (one document can belong to more topics), what is the name of the topic and what is the relationship of the topic to other topics. The main relationship "subtopic-of" is automatically induced when subtopics are added to the ontology as described in the previous section. The user can control all the relations between topics by adding, removing, directing and naming the relations.

Here the system can provide help on more levels:

- The system automatically assigns the documents to a topic when it is added to the ontology.
- The system helps by providing the keywords describing the topic using the methods described in Section 3. This can assist user when naming the topic.
- The system computes the cosine similarity between each document from the corpus and the centroid of the topic. This information can assist the user when searching for documents related to the topic. The similarity is shown on the list of documents next to the document name and the graph of similarities is plotted next to the list. This can be very practical when searching for outliers inside the concepts or for the documents that are not in the concepts but should be in considering their content.
- The system also computes similarities between the selected topic and all the other topics from the ontology. For the similarity measure between two topics it uses either the cosine similarity between their centroid vectors or the intersection between their documents.

5 OntoGen in Practice

In the previous sections we described the main components of the system and the text-mining techniques behind them. Here we will show how do these components

combine in a sample task of building a topic ontology and present users' feedback from cases studies which used OntoGen.

5.1 An Example Topic Ontology

In this section we will show example of a topic ontology constructed from 7177 company descriptions taken from Yahoo! Finance[2]. Each company is described with one paragraph of text. A typical description taken from Yahoo! would look as follows:

> Yahoo! Inc. is a provider of Internet products and services to consumers and businesses through the Yahoo! Network, its worldwide network of online properties. The Company's properties and services for consumers and businesses reside in four areas: Search and Marketplace, Information and Content, Communications and Consumer Services and Affiliate Services...

Using OntoGen on this descriptions one can in very little time (in our case just around 15 minutes!) create an ontology of areas that companies from Yahoo! Finance cover. Companies are also automatically positioned inside this ontology. The whole ontology generated with OntoGen is show in Figure 2 and zoom into part of the topic hierarchy depicted in Figure 3. In constructions of this topic ontology all the elements of OntoGen were used and the SVM keyword extraction method was shown to be very useful when naming topics that are far from the root. We kept most of the suggestions for topics and many of them were refined with help of our visualization for the outlier detection (see the bottom right part of Figure 1). Also, relation management was found very useful since the automatically discovered relations are not always optimal. By spending more time this ontology could be further developed to cover the ares in more details.

5.2 Case Studies and Users' Feedback

The presented system OntoGen was used for semi-automatic ontology construction in several case studies for modeling topic ontologies of the following domains:

- legal judgements in "Spanish legal case study" inside European project SEKT,
- virtual organizations inside inside European project ECOLEAD and
- news articles published by Slovene Press Agency (STA).

In all the three cases, the users were in fact domain experts, knowledgable about the domain but had little experience in knowledge engineering and practically no experience in machine learning. They were fast at learning how to use the system and were in general very pleased with its performance and the amount of time they needed to derive a desirable results. The domain experts were also satisfied with the final topic ontologies constructed for the all three cases.

[2] `http://finance.yahoo.com/`

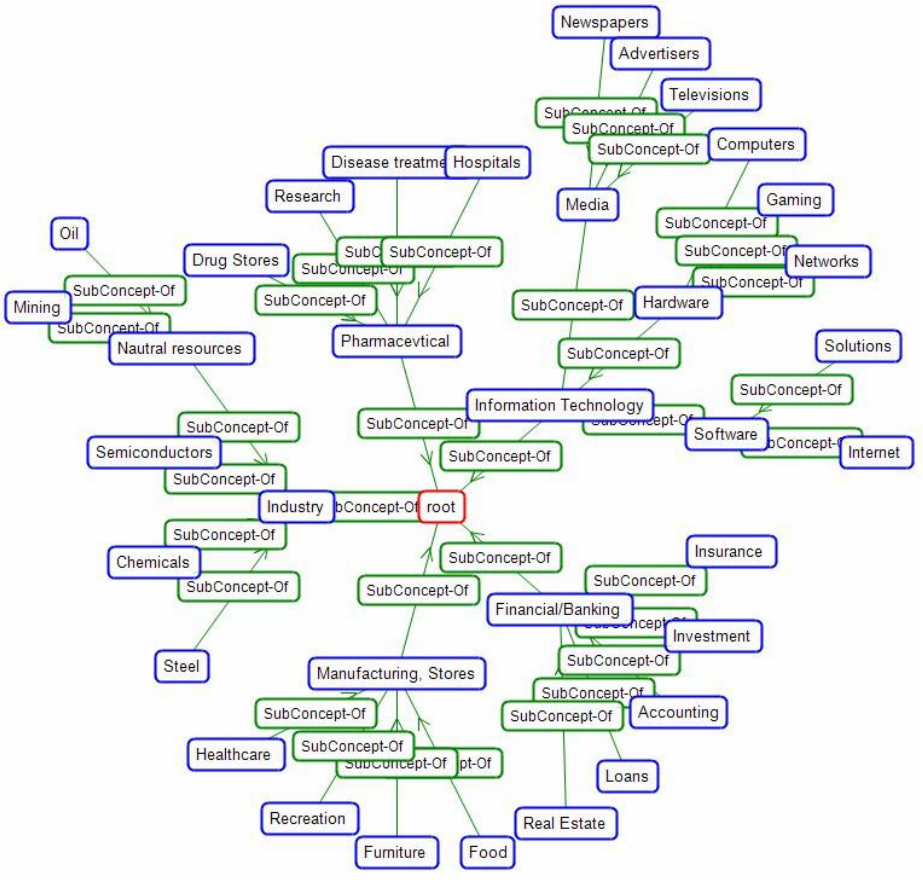


Fig. 2. Topic Ontology constructed from company descriptions. The top node of topic ontology is located in the center of the figure.

Many of the comments we got as a feedback from the users were related to the user interface of the system. For illustration, we list here the most interesting comments:

- no *undo* function,
- too little information is presented about suggested topics,
- editing of document membership for a specific topic is unclear and
- more interactive topic ontology visualization (folding, zooming).

Some other comments were more closely related to the topic suggestion and keyword extraction methods:

- "I would like to mark a keyword *not relevant* so the system will ignore it when generating suggestions?"
- "I know the name of the topic I would like to add to the topic ontology but the system does not find it."

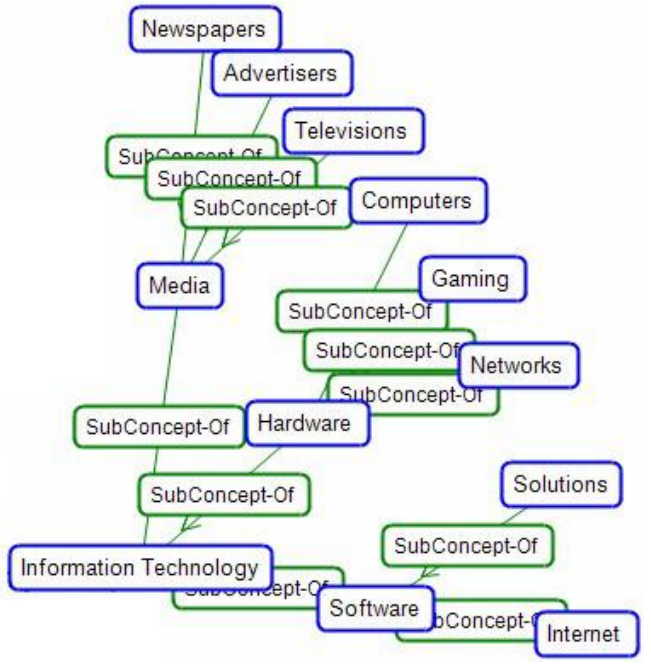


Fig. 3. Zoom in to *Information Technology*part of Yahoo! Finance topic ontology

These comments show the limits of the suggestion methods currently included in the system and they were of great help to use when deciding what other text-mining methods to include in the future versions of the system.

We found the comments from the users to be very informative and constructive and most of them will be implemented in the next versions of OntoGen.

6 Future Work

Currently we are working in two directions of adding functionality to OntoGen. The version presented in this paper can only help at discovering the topics but has no support for identification and naming of relations. The idea here is to use machine learning and text mining to discover possible relations between topics. The second direction is to include methods for incorporating background knowledge into the topic discovery algorithms [9]. This would enable building of different ontologies based on the same data. For example, the same document-database in a company may be viewed differently by marketing, management, and technical staff.

Another possible direction would be making the whole process more automatic and reduce the need for user interaction. This involves things like calculating the quality of topics suggested by the system, more automated discovery of the optimal number of topics, improved support for annotating the documents with the topics, discovering different kinds of relations between topics etc.

7 Conclusions and Discussion

In this paper we presented our approach to the semi-automatic construction of topic ontologies. In the first part of the paper we presented text mining techniques we used: two methods for discovering topics within the corpus, LSI and K-Means clustering, and two methods for extracting keywords. In the second part we showed how we integrated all these methods into an interactive system for constructing topic ontologies. The system was successfully tested and used in three case studies with very satisfactory results both in terms of final results and the feedback we got from the end-users.

Even though the system was primarily designed for constructing topic ontologies it can be generalized for other types of ontologies where the instances can be described by some relevant features. In case of topic ontologies the instances are documents which are described by words as features, but it might as well be users described by products they bought or movies they saw, images described by SIFT features, etc. Clustering can still be used as a method for discovering concepts but naming the concepts can be little more trickier for cases when features are harder to understand and are not words (for example, SIFT features used). In that cases methods for keyword extraction presented in this paper would not be sufficient.

Acknowledgments

This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT Semantically Enabled Knowledge Technologies (IST-1-506826-IP) and PASCAL Network of Excellence (IST-2002-506778). This publication only reflects the authors' views.

References

1. Agirre, E., Ansa, O., Hovy, E., Martinez. D. *Enriching Very Large Ontologies Using the WWW.* In Proceedings of the Ontology Learning Workshop, The 14th European Conference on Artificial Inteligence (ECAI), Berlin, Germany, 2000.
2. Bisson, G., Nedellec, C., Canamero L. *Designing clustering methods for ontology building: The MoK workbench.* In Proceedings of the Ontology Learning Workshop, The 14th European Conference on Artificial Inteligence (ECAI), Berlin, Germany, 2000.
3. Brank, J., Grobelnik, M., Milic-Frayling, N., Mladenic, D. *Feature selection using support vector machines.* In Proceedings of the 3rd International Conference on Data Mining Methods and Databases for Engineering, Finance, and Other Fields, Bologna, Italy, 2002.
4. Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S.: *Learning Taxonomic Relations from Heterogeneous Evidence.* In Proceedings of the Ontology Learning and Population Workshop, The 16th European Conference on Artificial Inteligence (ECAI), Valenci, Spain, 2004.

5. Deerwester, S., Dumais, S., Furnas, G., Landuer, T., Harshman, R.: *Indexing by Latent Semantic Analysis*. Journal of the American Society of Information Science, vol. 41, no. 6, 391-407, 1990.
6. Douglas B. Lenat R. V. Guha: *Building Large Knowledge-Based Systems* Addison Wesley, Reading, Massachusetts, 1990.
7. Lpez, M.F. *Overview of the methodologies for building ontologies*. In Proceedings of the Ontologies and Problem-Solving Methods Workshop, The 16th International Joint Conference on Artificial Inteligence (IJCAI), Stockholm, Sweden, 1999.
8. Fortuna, B., Grobelnik, M., Mladenic, D. *Visualization of text document corpus*. Informatica, vol. 29, 497-502, 2005.
9. Fortuna, B., Grobelnik, M., Mladenic, D. *Background Knowledge for Ontology Construction*. Poster at 16th International World Wide Web Conference (WWW2006), Edinburgh, Scotland, 2006.
10. Grobelnik, M., And Mladenic, D. *Efficient visualization of large text corpora*. In Proceedings of the 17th TELRI seminar, Dubrovnik, Croatia, 2002.
11. Heyer, G., Luter, M., Quasthoff, U., Wittig, T., Wolff, C. *Learning Relations using Collocations*. In Proceedings of Workshop on Ontology Learning, The 17th International Joint Conference on Artificial Inteligence (IJCAI), Seattle, USA, 2001.
12. Jain, A. K., Murty, M. N., Flynn, P. J. *Data Clustering: A Review*. ACM Computing Surveys, vol 31. no. 3, 264-323, 1999.
13. Joachims, T. *Making large-scale svm learning practical*. In Scholkopf, B., Burges, C., Smola, A., Advances in Kernel Methods: Support Vector Machines, MIT Press, Cambridge, MA, 1998.
14. Leskovec, J., Grobelnik, M., Milic-Frayling, N. *Learning Semantic Graph Mapping for Document Summarization*. In Proceedings of Workshop on Knowledge Discovery and Ontologies, 15th European Conference on Machine Learning (ECML) and 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Pisa, Italy, 2004.
15. Maedche, A., Staab, S. *Discovering conceptual relations from text*. In The 14th European Conference on Artificial Inteligence (ECAI), 321-325, Berlin, Germany, 2000.
16. Reinberger, M-L., Spyns, P. *Discovering Knowledge in Texts for the learning of DOGMA-inspired ontologies*. In Proceedings of the Ontology Learning and Population Workshop, The 16th European Conference on Artificial Inteligence (ECAI), Valenci, Spain, 2004.
17. Salton, G. *Developments in Automatic Text Retrieval*. Science, vol. 253, pages 974-979, 1991.
18. Steinbach, M., Karypis, G., Kumar, V.): *A comparison of document clustering techniques*. In Proceedings of KDD Workshop on Text Mining, 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Boston, USA, 2000.
19. Uschold, M. *Towards a Methodology for Building Ontologies*. Workshop on Basic Ontological Issues in Knowledge Sharing, The 14th International Joint Conference on Artificial Inteligence (IJCAI), Motnreal, Canada, 1995.

Evaluation of Ontology Enhancement Tools*

Myra Spiliopoulou[1], Markus Schaal[2,**], Roland M. Müller[1], and Marko Brunzel[1]

[1] Otto-von-Guericke-Universität Magdeburg
[2] Bilkent University, Ankara

Abstract. Mining algorithms can enhance the task of ontology establishment but methods are needed to assess the quality of their findings. Ontology establishment is a long-term interactive process, so it is important to evaluate the contribution of a mining tool at an early phase of this process so that only appropriate tools are used in later phases. We propose a method for the evaluation of such tools on their impact on ontology enhancement. We model impact as quality perceived by the expert and as statistical quality computed by an objective function. We further provide a mechanism that juxtaposes the two forms of quality. We have applied our method on an ontology enhancement tool and gained some interesting insights on the interplay between perceived impact and statistical quality.

1 Introduction

The manual establishment of ontologies is an intriguing and resource-consuming task. Efforts are made to enhance this process by unsupervised learning methods. However, as pointed out in [11], the semantic richness and diversity of corpora does not lend itself to full automation, so that the involvement of a domain expert becomes necessary. Hence, unsupervised tools undertake the role of providing useful suggestions, whereupon the quality of their contributions must be evaluated. Since ontology enhancement is a long-term process involving multiple corpora and possibly multiple iterations over the same corpus, this evaluation should be done at an early step, so that only appropriate tools are considered in later steps. In this study, we propose a method for the early evaluation of clustering tools that suggest correlated concepts for ontology enhancement.

Our method has two aspects: First, it evaluates the *impact* of the tool's suggestions as *perceived* by the domain expert. Second, it juxtaposes the *objective quality* of these suggestions to the perceived impact. While the objective quality refers to the statistical properties of the discovered patterns, such as the confidence of a rule or the homogeneity of a cluster, the impact is reflected in the ultimate decision of the expert to include the suggested pattern in the ontology or not. The juxtaposition of the objective, tool-internal notion of quality to the quality perceived by the expert indicates whether the tool and its quality measures will be helpful in further steps of the ontology establishment process.

In the next section, we discuss related work on the evaluation of unsupervised learning tools. In section 3 we describe our method for impact evaluation by the domain expert, followed by the method juxtaposing impact and statistical quality. In section 4, we briefly present the tool we have used as experimentation example. Section 5 describes our experiments and acquired insights. The last section concludes our study.

* Work partially funded under the EU Contract IST-2001-39023 Parmenides.
** Work done while with the Otto-von-Guericke-Universität Magdeburg.

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 132–146, 2006.

2 Related Work

Ontology learning tools as proposed in [1,2,4,6,10,8,14] serve different purposes. Many of them propose objects (concepts and relationships) that are found to be supported by a document collection relevant to the application at hand. We concentrate on tools that enhance an existing ontology by proposing (a) new concepts to be inserted in it and (b) relationships among existing concepts.

Usually, an ontology enhancement tool has an inherent quality assessment mechanism that rejects patterns according to some scheme. For tools based on association rules' discovery, quality assessment is often based on interestingness and unexpectedness, while cluster quality is often based on homogeneity or compactness. A rich collection of criteria for the statistical evaluation of unsupervised learners has appeared in [16]. It contains valuable criteria for the assesment of cluster quality, many of them based on indexes of cluster homogeneity. More oriented towards the needs of text clustering are the criteria considered in [15], in which a correlation between some cluster homogeneity indexes and the F-measure is identified when experimenting upon a gold standard. However, application-specific ontology learning cannot rely on gold standards developed for different applications. Moreover, cluster homogeneity does not guarantee or imply that the cluster labels will also be interesting to the domain expert.

Evaluation from the viewpoint of ontology learning is more challenging. Holsapple and Joshi proposed an evaluation method for collaborative manual ontology engineering, in which each suggestion made by one expert is evaluated by at least another expert [7]. Hence, good suggestions are those that enjoy the approval of multiple experts. While this is reasonable for ontology engineering among human experts, it cannot be transferred to non-human experts: Agreement among several ontology learners does not necessarily imply that human experts will find their suggestions useful, since ontology learners are based more on statistics than on background knowledge and expert insight.

The ECAI 2004 workshop on "Ontology Learning and Population" concentrated on the subject of "Evaluation of Text-based Methods in the Semantic Web and Knowledge Discovery Life Cycle" [1]. Faatz and Steinmetz proposed an elegant formalization of "ontology enrichment", followed by a method for automated evaluation on the basis of precision and recall [3], i.e. with respect to gold standards. The selection of those measures is in accordance with the task of evaluation *for algorithmic tuning*: The authors state that "only automatic evaluations of ontology enrichment meet the requirements of algorithmic tuning" and that "the automatization has to be aware of the task specific semantic direction, to which an ontology should evolve" [3]. In our study, we pursue a different goal: We want to assist an expert in deciding on the appropriateness of the tool rather than tune any tool. Moreover, we deliver a procedure that decides whether algorithmic tuning should be made or rather avoided as incompatible to the preferences/intuition of the expert.

Porzel and Malaka consider task-oriented evaluation of ontologies [13]. The process creating an ontology is not specified explicitly, but (semi-)automated processes seem to be permissible; a tool could be evaluated on the quality of the ontology it produces. The authors consider evaluation only with respect to a predefined task, since ontologies

[1] http://olp.dfki.de/ecai04/cfp.htm, accessed at July 26, 2005.

are indeed built to serve specific tasks. Their evaluation method is based on error-rates, namely superfluous, ambiguous or missing concepts with respect to the task [13]. For our objective of appropriateness evaluation for tools, this approach has some shortcomings. First, it evaluates whole ontologies, while we are interested in the stepwise enhancement of a preliminary ontology. Second, evaluation on the basis of error rates requires a gold standard tailored to the anticipated task. The establishment of such a standard is quite counterintuitive from the viewpoint of a domain expert that needs a tool to enhance an ontology with concepts she does not know in advance.

Kavalec and Svatek propose a method for the evaluation of relation labels, i.e. combinations of terms proposed by a text mining tool to a human expert [9]. According to this method, the expert proposes labels for identified relations and then the tool attempts to re-discover those labels (or synonyms of the terms in them) by mining the text collection. By nature, this approach is appropriate when evaluating a tool on the basis of existing, a priori known relations in a known ontology but less so when evaluating the appropriateness of a tool in expanding an ontology in unknown ways according to the demands of a given expert.

Navigli et al proposed an evaluation method for the OntoLearn system, encompassing a quantitative evaluation towards specific corpora and a qualitative evaluation by multiple domain experts [12]: Quantitative evaluation for the term extraction algorithm, the ontology learning algorithm and the semantic annotation algorithm was performed on predefined corpora which served as gold standards. While this type of evaluation allows for conclusions about the robustness of one tool or the relative performance of multiple tools, it does not allow for generalizations on the usefulness of a given tool to a given expert for the enhancement of a given ontology from a given document collection.

The qualitative evaluation proposed in [12] was based on a questionnaire, in which experts assessed the quality of the definitions of the concepts discovered by OntoLearn: The complex concepts found by the OntoLearn rules were combined with concept definitions from WordNet. The experts were then asked to rate the glosses thus generated as unacceptable, helpful or fully acceptable. This is closer to our one-expert evaluation. However, we do not consider concept definitions, because (a) an appropriate definition provider may or may not be available – the WordNet is not appropriate for specialized domains, (b) the interpretation of a complex concept is left to the expert and (c) a small or medium enterprise intending to enhance an ontology is more likely to dedicate one domain expert to this task rather than 10 or 25 experts. So, the approach is not applicable for *providing assistance* to *one* expert. Further, the appropriateness of the selected corpus was taken for granted; in our approach, this assumption is being put to test.

A method for the generation and evaluation of suggestions towards an ontology user is proposed in [5]. The authors propose a recommendation engine that explores the activities of multiple users, who expand their personal ontologies from a shared basic ontology and suggest metrics for the evaluation of the engine's suggestions. This approach is appropriate when ontologies are built collaboratively by people, since the actions of one person may be helpful to others. However, the metrics do not apply for the actions (suggestions) of one tool towards one domain expert.

3 A Posteriori Impact Evaluation for Ontology Enhancement

Our method evaluates the impact of text miners on the task of ontology enhancement. A text miner processes a text collection and suggests semantics that expand the ontology. These may be terms like "hurricane" or "hurricane warning", term groups that form a new concept or a relation among concepts, e.g. "hurricane warning area", or named relations like "expected within". We refer to them as *"concept constellations"* and focus on the evaluation of the process discovering them. We focus on tools for text clustering and labeling, but our method can be easily extended for association rules' discovery.

3.1 Objectives

We observe ontology enhancement as an iterative process performed by a text mining tool that is applied on an application-specific document collection. The initial input is a preliminary ontology to be enhanced with help of each collection. The final output should be an enriched ontology that is "complete towards the collection", in the sense that the collection cannot contribute new concept constellations to it. More specifically:

- The original input ontology contains a hierarchy or multiple hierarchies of concepts, that may be further connected with horizontal (labeled or unlabeled) relations.
- A text mining tool attempts to enrich the ontology by processing a document collection and identifying semantically correlated concepts. Such correlations are assumed to manifest themselves as concepts that appear frequently together, e.g. as collocates (physically proximal concepts in texts) or as groups of concepts that characterize a cluster of documents. As already noted, we concentrate on text clustering tools. We use the term "concept constellation" for a group of correlated concepts that is returned by the tool as label of a text cluster.
- The correlations among the concepts are used to enrich the ontology. The concepts themselves are already in the ontology, so the enrichment can take two forms, namely the insertion of horizontal relations among the involved concepts and the definition of a new concept that summarizes the correlated concepts.
- The tool finds these concept constellations by mining a document collection.
- The ontolgoy is enriched in many iterations. In each one, the input is the updated ontology. The iterative process ends when no further enrichment can be performed.

Our evaluation method is intended for the first iteration of this process and should answer the following questions:

1. Is the tool appropriate for the enhancement of *this* ontology – on *this collection*?
2. Is the collection appropriate for the enhancement of the ontology – with this tool?
3. Are the tool's quality evaluation functions aligned to the demands of *this* expert?

The motivation of the first question is that a tool may perform well for one collection and poorly for another. A collection can itself be inappropriate for the enhancement of the specific ontology and indeed for opposite reasons: At the one end, the collection may be only marginally relevant, as would be a document collection on outdoor sport

for an ontology on hurricanes. At the other end, the collection may have already served as inspiration for the ontology, whereupon it cannot be used any more to enhance the ontology further.

The last question stresses the subjectivity of the ontology enhancement process. This subjectivity cannot be expelled altogether. However, by modeling the evaluation process on the basis of those three questions, we ensure that the implicit preferences of the expert are partially explicated when dealing with the first two questions. Those preferences that remain tacit are reflected in the outcome of the last question.

We present the evaluation model with respect to the first two questions hereafter and focus on the third question in Section 3.4.

3.2 Perceived Quality as Relevance + Appropriateness

We evaluate the tool's impact on ontology enhancement as *perceived* by the ontology expert. We use two criteria, "relevance to the application domain" and "appropriateness for the ontology O", where D stands for the collection *as representative of the application Domain*. To this purpose, we define two functions $R(D)$ and $A(O, D)$: They are used to measure the relevance of a collection D, resp. its appropriateness for enhancing O within the application domain.

Relevance to the Application Domain. The ontology enhancement is assumed to take place in the context of an application domain and that the collection is representative of that domain. For this criterion, the domain expert is asked to characterize each suggestion (concept constellation) made by the tool as relevant or irrelevant to that domain, *independently of whether she considers the suggestion as appropriate for the ontology*.

The term "relevance" is known to be very subjective. However, the intention of this criterion is not to assess the relevance of the individual suggestions but rather the appropriateness of the tool and of the collection for the application domain. In particular, consider the task of discovering correlated concepts in the following excerpt from the National Hurricane Center at `www.noaa.com`:

```
A HURRICANE OR TROPICAL STORM WARNING MEANS THAT HURRICANE OR
TROPICAL STORM CONDITIONS ... RESPECTIVELY ... ARE EXPECTED
WITHIN THE WARNING AREA WITHIN THE NEXT 24 HOURS. PREPARATIONS
TO PROTECT LIFE AND PROPERTY SHOULD BE RUSHED TO COMPLETION IN
THE HURRICANE WARNING AREA.
```

For the application area of extreme weather warnings, a tool applied on the text collection might suggest the following concepts / constellations, listed here in alphabetical order: (I) "storm, tropical, warning", "area, hurricane, warning", "preparations, protect", (II) "hurricane", "storm", (III) "are, expected", "area". Note that we do not check whether the tool can assess that e.g. "hurricane warning area" is one or two concepts.

- Suggestions of type I are relevant. If most suggestions are of this type, then the tool is appropriate for the collection.
- Suggestions of type III are irrelevant and indicate that the tool cannot find relevant concept constellations upon this collection. If most suggestions are of this type, it should be checked whether the collection itself is appropriate. If yes, then the tool is not appropriate for it.

– Type II suggestions are more challenging. An expert may reject the suggestion "hurricane" as uninformative for an application domain on hurricanes. However, with respect to our criterion, such suggestions should be marked as relevant: Informativeness and appropriateness for the ontology are addressed by our next criterion.

Appropriateness for the Ontology. The Appropriateness criterion $A(O, D)$ refers to the expansion of ontology O for the application domain D. It builds upon the relevance criterion $R(D)$: only relevant concept constellations are considered. For a relevant concept constellation $Y = Y_1, \ldots, Y_m$, the following cases may occur:

– Y is already in the ontology. Then it should be rejected as inappropriate.
– Y contains some concepts that are appropriate for the ontology, either as individual concepts or as a group. Then Y should be accepted; each appropriate concept/group should be named.
– Y contains no concept that is appropriate for the ontology. It should be rejected.

According to this scheme, a concept constellation may contribute one or more concepts to the ontology. Hence, $A(O, D)$ delivers two lists of results: $A(O, D) = \{S, S_+\}$, where $S \subseteq R(D)$ is the set of accepted concept constellations and S_+ is the set of concept groups appropriate for the ontology.

We use the result $A(O, D).S$ to assess the appropriateness of the tool for further iterations in the ontology enhancement process. The result $A(O, D).S_+$ is used in 3.4, where we juxtapose the quality criteria of the tool to the impact perceived by the expert.

3.3 Combining Relevance and Appropriateness Ratings

Let $T(D)$ be the set of concept constellations suggested by the tool T for the application domain. We combine the results on relevance $R(D) \subseteq T(D)$ and appropriateness for the ontology $A(O, D).S$ to figure out whether the tool T should be further used for the enhancement of the ontology on domain D, whereupon we consider the collection already analyzed as representative for domain D. The following cases may occur:

– The ratio $\frac{|R(D)|}{|T(D)|}$ is close to zero.
Then, the tool is not appropriate for this collection and thus for the domain.
– The ratio $\frac{|R(D)|}{|T(D)|}$ is closer to one and the ratio $\frac{|A(O,D).S|}{|R(D)|}$ is close to zero.
Then, the tool is capable of analyzing documents in the application domain but the collection does not deliver informative concept constellations for the ontology. This may be due to the tool or to the relationship between ontology and collection. To exclude the latter case, the domain expert should again verify the appropriateness of this collection *for ontology enhancement*: If all concepts in the collection are already in the ontology, the collection is still relevant but cannot enrich the ontology any more. Hence, the tool should be tested upon another representative collection.
– Both ratios are closer to one than to zero.
Then, the tool is able to contribute to ontology enhancement for this collection and is thus appropriate for the application domain.

By this procedure, we can assess whether a given tool should be further used for the gradual enhancement of the ontology. For a more *effective* ontology enhancement process, it is also reasonable to know to which extent the tool's suggestions can be trusted without close inspection. This would be the case if the enhancements proposed by the tool fit to the expectations of the human expert (cf. Question 3 in Section 3.1). To this purpose, we juxtapose the evaluation by the expert to the internal quality evaluation by the tool. Obviously, this juxtaposition is only possible for tools that disclose the values assigned to their suggestions by their internal evaluation criteria. For tools delivering only unranked suggestions, no juxtaposition is possible.

3.4 Juxtaposition of Statistical and Perceived Quality

Each (text clustering) tool has some internal or external criterion for the rejection of potentially poor patterns and the maintenance, respectively further exploitation, of good patterns. The results of any clustering algorithm encompass both good and less good clusters, whereby goodness is often measured in terms of compactness, homogeneity, informativeness etc [15,16]. We name such criteria "statistical quality criteria".

Towards our objective of ontology enhancement, we say that a statistical quality criterion $SQ()$ *"is aligned to the perceived quality"* when the likelihood that the domain expert considers a concept group as appropriate for the ontology increases (resp. decreases) with the statistical quality of the cluster with respect to that criterion.

As basis for the statistical quality, let $SQ()$ be a statistical quality criterion that assigns to each cluster generated by T a value. Without loss of generality, we assume that the range of these values is $[0, 1]$ and that 1 is the best value. As basis for the perceived quality, we consider the concept groups characterized by the domain expert as appropriate for the ontology, i.e. the set $A(O, D).S_+$ defined in 3.2.

Associating Concept Groups and Constellations with Clusters. To compare the perceived with the statistical quality of the concept groups and constellations, we compute the distribution of statistical quality values for the concept groups accepted by the expert and for the concept constellations rejected by her.

Since an accepted concept group, i.e. an element of $A(O, D).S_+$ may appear in more than one concept constellations, it can be supported by one or more clusters of the clustering $T(D)$ generated by the tool and these clusters may be of different statistical quality. Hence, we associate each concept group $x \in A(O, D).S_+$ to the best among the clusters supporting it, C_x and then to the quality value of this cluster $SQ(C_x)$. We denote the set of pairs $(x, SQ(C_x))$ thus computed as $expertApproved$.

Similarly, we associate each rejected concept constellation $x \in T(D) \setminus A(O, D).S$ to the cluster C_x from which it was derived. Differently from the concept groups which may be supported by several clusters, a concept constellation corresponds to exactly one cluster, so the assignment is trivial. We denote the set of pairs $(x, SQ(C_x))$ thus computed as $expertRejected$.

In Table 1, we show the algorithm that computes the two sets $expertApproved$ and $expertRejected$. For each concept group $x \in A(O, D).S_+$ all clusters that deliver concept constellations containing x. It selects among them the cluster with the highest quality value according to $SQ()$ and associates x to this $maxSQ(x)$ (lines 3-7). The filling of the two sets of value pairs in the lines 8, 11 is straightforward.

Table 1. Associating each concept group to the best quality cluster

```
1  expertApproved:=expertRejected=∅
2  For each concept group x in A(O,D).S+
3     maxSQ := 0
4     For each cluster C in T(D) that supports x
5        if maxSQ less than SQ(C)
6           then maxSQ := SQ(C)
7     Endfor
8     expertApproved:=expertApproved∪{(x,maxSQ)}
9  Endfor
10 For each concept constellation x in T(D)\A(O,D).S
11    expertRejected:=expertRejected∪{(x,SQ(C_x))}
12 Endfor
```

Comparing Distributions. The two sets $expertApproved$ and $expertRejected$ can be mapped into distributions of statistical quality values for the accepted, resp. rejected clusters. We denote these distributions as dA and dR respectively. To check whether statistical quality and perceived quality are aligned, we should compare those distributions. However, since the concrete distributions are not known, we can either (a) derive histograms hA and hR for them by partitioning the valuerange of $SQ()$ into k intervals for some k or (b) compute the mean and standard deviation of each dataset. Then, the form of the histograms or the values of the means are compared. For the comparison of histograms, we consider the cases depicted in Table 2.

By this juxtaposition we can assess whether a statistical quality criterion used by the tool is aligned to the implicit perceived quality function of the domain expert. If some criteria are aligned, they should take priority over misaligned ones in subsequent ontology enhancement steps. Even if all criteria are misaligned, the tool can still be

Table 2. Comparison of histograms - four cases

1. Both histograms are unimodal, hA is shifted towards the best quality value for $SQ()$, while hR is shifted towards the worst value.
   This is the best case: The likelihood that a cluster contributes to ontology enhancement increases with its quality and vice versa. $SQ()$ is aligned to perceived quality.
2. Both histograms are unimodal, hR is shifted towards the best value and hA is shifted towards the worst value.
   This is the second best case. The statistical quality criterion is *consistently* counterproductive. One might reasonably argue that this $SQ()$ is a poor criterion, but it is also true that $1-SQ()$ is aligned to the perceived quality and is thus very useful.
3. The two histograms have the same shape and are shifted in the same direction.
   Then the likelihood of having a good cluster accepted or rejected by the expert is the same as for a bad cluster. Thus, $SQ()$ is misaligned to the perceived quality.
4. No pattern can be recognized. Then $SQ()$ is misaligned to the perceived quality.

used. However, it should then deliver to the domain expert the poor quality clusters as well, since she may find useful information in them.

A comparison based on histograms depends on the selected number of intervals k and on the specific partitioning of the valuerange of $SQ()$. An alternative, simpler approach would be to compute the proximity of the median to the best, resp. worst value of $SQ()$: Similarly to the comparison of the histograms, if the median of $expertApproved$ is close to the best value of $SQ()$ and the median of $expertRejected$ is close to the worst value, then $SQ()$ is aligned; if the reverse is the case, then $SQ()$ is consistently counterproductive. Otherwise, $SQ()$ is misaligned.

4 An Example Tool and Its Quality Evaluation Criteria

As a proof of concept, we have applied our evaluation method upon the tool "RELFIN Learner" [14]. We describe RELFIN and its internal quality evaluation criteria below, mostly based on [14]. We stress that RELFIN is only an example: Our method can be applied on arbitrary tools that suggest concepts for ontology enhancement. Obviously, the juxtaposition to a tool's statistical quality is only feasible if the tool reports its quality assessment values as required in 3.4.

RELFIN is a text clustering algorithm using Bisecting-K-means as its clustering core and a mechanism for cluster evaluation and labeling. RELFIN discovers new concepts as single terms or groups of terms characterizing a cluster of text units. These concepts, resp. concept constellations can be used to expand the existing ontology, to semantically tag the corresponding text units in the documents or to do both. RELFIN can take as input both concepts from an initial, rudimentary ontology and with additional terms it extracts automatically from the collection. Accordingly, its suggestions are new concepts consisting of terms in the collection and constellations consisting of terms from either the ontology or the collection. The labels / concept constellations suggested by RELFIN should be appropriate as semantic markup on the text fragments. This is reflected in the quality criteria of RELFIN.

4.1 Definitions

A *text unit* is an arbitrary text fragment extracted by a linguistic tool, e.g. by a sentence-splitter; it is usually a paragraph or a sentence. Text units are composed of terms. For our purposes, a *text collection* $\mathcal{A}$ is a set of text units.

A term is a textual representation of a *concept*. A *feature space* $\mathcal{F}$ consists of concepts from the existing ontology, terms extracted from the collection by some statistical method or both. We assume a feature space with d dimensions and a *vectorization* $\mathcal{X}$ in which each text unit i is represented as vector of TFxIDF weights $x_i = (x_{i1}, \ldots, x_{id})$. Obviously, concepts of the ontology that do not appear in the collection are ignored.

Given is a *clustering scheme* or *clusterer* $\mathcal{C}$. For a cluster $C \in \mathcal{C}$, we compute the in-cluster-support of each feature $f \in \mathcal{F}$ as

$$ics(f, C) = \frac{|\{x \in C | x_f \neq 0\}|}{|C|} \quad (1)$$

Definition 1 (Cluster Label). *Let $C \in \mathcal{C}$ be a cluster over the text collection $\mathcal{A}$ for the feature space $\mathcal{F}$. The* label *of C $label(C)$ is the set of features $\{f \in \mathcal{F} | ics(f, C) \geq \tau_{ics}\}$ for some threshold τ_{ics}.*

A feature satisfying the threshold constraint for a cluster C is a *frequent feature* for C.

4.2 Quality Measures

A label might be specified for any cluster. To restrict labeling to good clusters only, we use one criterion on cluster compactness and one on feature support inside clusters.

Definition 2 (Average distance from centroid). *Let $C \in \mathcal{C}$ be a cluster over the text collection $\mathcal{A}$ for the feature space $\mathcal{F}$ and let $d()$ be the distance function for cluster separation. The average intra-cluster distance from the centroid is defined as $avgc(C) = \frac{\sum_{x \in C} d(x, centroid(C))}{|C|}$, whereupon lower values are better.*

Definition 3 (Residue). *Let $C \in \mathcal{C}$ be a cluster and let τ_{ics} be the in-cluster support threshold for the cluster label. Then, the "residue" of C is the relative in-cluster support for infrequent features:*

$$residue(C, \tau_{ics}) = \frac{\sum_{f \in \mathcal{F} \setminus label(C)} ics(f, C)}{\sum_{f \in \mathcal{F}} ics(f, C)} \tag{2}$$

The residue criterion serves the goal of using cluster labels for semantic markup. Consider text units that support the features X and Y and text units that support Y and Z. If the algorithm assigns them to the same cluster, then both pairs of features can be frequent, depending on the threshold τ_{ics}. A concept group "X,Y,Z" may well be of interest for ontology enhancement, but it is less appropriate as semantic tag. We allow for low τ_{ics} values, so that such constellations can be generated. At the same time, the residue criterion favours clusters dominated by a few frequent features shared by most cluster members, while all other features are very rare (values close to zero are best).

5 Experiments

We performed an experiment on ontology enhancement involving a domain expert who used the RELFIN Learner for the enhancement of an existing ontology. The expert's goal was to assess usability of the tool. The complete usability test is beyond the scope of this study, so we concentrate only on the impact assessment criteria used in the test. The juxtaposition to the statistical criteria of the tool was not part of the usability test.

5.1 The Case Study for Ontology Enhancement

Our method expects a well-defined application domain. This was guaranteed by a predefined case study with a given initial ontology on biotechnology watch and two domain-relevant collections of business news documents. We used a subcollection of BZWire news (from 1.1.2004 to 15.3.2004), containing 1554 documents. The vectorization process resulted in 11,136 text fragments.

The feature space consisted of 70 concepts from the initial ontology and 230 terms extracted from the collection. These terms were derived automatically as being more frequent for the collection than for a reference general purpose corpus. The target number of clusters was set to 60 and the in-cluster-support threshold for cluster labeling τ_{ics} was set to 0.2. Setting τ_{ics} to such a rather low value has turned to be helpful for our observations, because high values would reduce the set of suggestions considerably.

5.2 Evaluation on Relevance and Appropriateness

RELFIN delivered 60 clusters of varying quality according to the tool's internal criteria. For the impact assessment by the domain expert, though, these criteria were switched off, so that all cluster labels subject to $\tau_{ics} = 0.2$ were shown to the domain expert. This implies that RELFIN suggested the labels of all 60 clusters, so that $|T(D)| = 60$.

The domain expert was asked to assess the relevance of each cluster label, i.e. constellation of frequent features. A label was relevant if it contained at least one relevant feature. The appropriateness of the features in relevant cluster labels was assessed next: The domain expert was asked whether NONE, ALL or SOME of the concepts in the relevant label were also appropriate. The answers were:

- *Relevance to the case study:* YES: 43, NO: 17 $|R(D)| = 43$
- *Appropriateness for the ontology:* NONE: 2, ALL: 4, SOME: 37 $|A(O, D).S| = 41$

We combined these values as described in 3.3. To compute $A(O, D).S_+$, we enumerated the concept groups in the labels characterized as SOME, using the following rules:

1. The expert saw a label with several concepts and named n concept groups that he considered appropriate for the ontology. Then, we counted n appropriate objects.
2. The expert found an appropriate concept or concept group and marked it in *all* cluster labels containing it. Then, we counted the appropriate object only once.
3. The domain expert saw a label "A,B,C,…", and wrote that "A,B" should be added to the ontology. Then, we counted one appropriate object only, even if the terms "A" and "B" did not belong to the ontology.
4. The expert saw a label of many concepts and marked them "ALL" as appropriate. This case occured 4 times. For three labels, we counted one appropriate object only, independently of the number of new concepts and possible combinations among them. For the 4th label, we counted two appropriate objects: the label as a whole and one specific term X. X belongs to a well-defined set of terms and the expert had encountered and accepted three further members of this set when evaluating other clusters. So we added this term, too.

In Table 3 we show the relevance and appropriateness ratios according to those rules. These ratios allow for an assessment (positive in this case) of the tool's appropriateness for further iterations. In the last rows, we have computed the average number of appropriate concept groups, as contributed by the RELFIN clusters. The last ratio is peculiar to RELFIN, which can exploit both concepts from the ontology and terms from the collection. The ratio says that 87% of the approved concept groups were not in the ontology. The remaining 23% are combinations of concepts from the ontology.

Table 3. Relevance and appropriateness ratios

Tool suggestions	$\lvert T(D)\rvert$	60
Relevance ratio	$\frac{\lvert R(D)\rvert}{\lvert T(D)\rvert}$	$43/60 \approx 0.72$
Appropriateness ratio	$\frac{\lvert A(O,D).S\rvert}{\lvert R(D)\rvert}$	$41/43 \approx 0.95$
Avg contribution of concept groups per relevant cluster		$62/43 \approx 1.44$
Avg contribution of concept groups per cluster		$62/60 \approx 1.03$
Contribution of the collection to the ontology		$54/62 \approx 0.87$

5.3 Impact Versus Statistical Quality

For the juxtaposition of the impact evaluation with the statistical quality criteria of RELFIN, we used the approach described in 3.4. Both criteria used by RELFIN range in the interval $[0, 1]$; 1 is the worst value and 0 is the best one. We have adjusted the generic procedure accordingly for the experiment.

In Table 4 we show the histograms for RELFIN. We have set the number of intervals to $k = 10$. However, we have noticed that all values of relevance according to Table 2 were in the intervals between 0.3 and 0.5 for the criterion "average distance from the centroid" $avgc$ and between 0.2 and 0.6 for the criterion "residue". Therefore, we have summarized the corresponding $SQ()$ values for the first two intervals into $[0, 0.2)$ and for the last intervals into $[0.5, 1)$ for the $avgc$ and into $[0.6, 1)$ for the residue.

Table 4. Quality values for approved vs rejected clusters

	Avg Distance to centroid					
	[0,0.2)	[0.2,0.3)	[0.3,0.4)	[0.4,0.5)	[0.5,1]	
Approved concept groups	2	7	19	27	6	
expertApproved clusters	2	5	12	17	7	
expertRejected clusters	1	1	1	4	10	
	Residue					
	[0,0.2)	[0.2,0.3)	[0.3,0.4)	[0.4,0.5)	[0.5,0.6)	[0.6,1]
Approved concept groups	0	2	6	16	12	25
expertApproved clusters	0	2	4	9	9	19
expertRejected clusters	1	3	4	0	3	6

For each criterion, the first row shows the distribution of cluster quality values for the approved concept groups. As pointed out in Section 3.4, a concept group may be supported by more than one clusters, from which the one with the highest quality is chosen (cf. Table 1). The second row shows the cluster quality values per interval for the approved clusters, i.e. for the set $expertApproved$. The third row shows the corresponding distribution for the clusters in $expertRejected$.

For the criterion $avgc()$, most values of hA (clusters in $expertApproved$) are in $[0.3, 0.5)$; a steep decrease occurs afterwards. For the hR (clusters in $expertRejected$),

most values are in $[0.5, 1)$. The median of $expertApproved$ is in the interval $[0.4, 0.5)$, the median of $expertRejected$ is larger than 0.5. These observations are indicatory of the first case in Table 2, hence the "average distance to the centroid" $avgc()$ is aligned to the expert's evaluation.

In the first row of the criterion "residue", we can see a modus in the interval $[0.4, 0.5)$. It is followed by a smaller modus in the next interval $[0.5, 0.6)$, which also contains the median. It must be stressed that the last interval is an aggregate; there is no modus there. The value distribution in the hA for the $expertApproved$ clusters is in the second row: The modus spans over the two intervals $[0.4, 0.5)$ and $[0.5, 0.6)$; the latter contains the median. However, the histogram of $expertRejected$ clusters has at least two modi, one before the interval $[0.4, 0.5)$ and at least one afterwards; this interval is itself empty. Hence, the likelihood of a cluster rejection is high both before and after this interval. So, we conclude that the criterion is misaligned.

One explanation of the misalignment of the residue is that the labels of clusters with higher residue contain more concepts. When the human expert identified appropriate concept groups for the ontology, he had more candidates to choose from. Those concept groups are not appropriate as semantic tags but this does not affect their appropriateness for the ontology. We consider this as indicatory for impact assessment: If a concept (group) appeals to the domain expert, i.e. is informative with respect to her background knowledge, she will approve it independently of its statistical support.

6 Conclusions

We have proposed a method that evaluates the appropriateness of text clustering tools for ontology enhancement on the basis of their suggestions to the domain expert. Our approach is intended as an instrument to help the domain expert decide at the beginning of the ontology enhancement process whether the tool is appropriate for further steps of this process. To this purpose, we combine subjective impact assessment with a more objective relevance test and we finally check whether the statistical evaluation instruments used by the tool are aligned to the subjective preferences of the expert. We have performed a first test of our method for a text clustering tool on the enhancement of the ontology of a real case study and we gained some rather interesting insights on the interplay of statistical "goodness" and subjective "appropriateness".

The juxtaposition of statistical quality and impact assessment might be observed as a classification task, where statistical criteria serve as predictors of impact. We intend to investigate this potential. We further plan to enhance the impact assessment with more elaborate criteria. Moreover, we want to evaluate further tools with our method: This implies conducting an experiment in which the expert works with multiple tools on the same corpus and the same basic ontology.

Acknowledgement. We would like to thank the domain expert Dr. Andreas Persidis of the company BIOVISTA for the impact evaluation and for many insightful comments on the expectations towards interactive tools used in ontology enhancement.

References

1. Philipp Cimiano, Steffen Staab, and Julien Tane. Automatic acquisition of taxonomies from text: Fca meets nlp. In *Proc. of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining*, pages 10–17, Cavtat, Croatia, Sept. 2003.
2. Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. SemTag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proc. of the 12th Int. World Wide Web Conf.*, pages 178–186, Budapest, Hungary, 2003. ACM Press.
3. Andreas Faatz and Ralf Steinmetz. Precision and recall for ontology enrichment. In *Proc. of ECAI-2004 Workshop on Ontology Learning and Population*, Valencia, Spain, Aug. 2004. http://olp.dfki.de/ecai04/cfp.htm, accessed at July 26, 2005.
4. David Faure and Claire Nédellec. Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM. In Dieter Fensel and Rudi Studer, editors, *Proc. of 11th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW'99)*, volume LNAI 1621, pages 329–334, Dagstuhl, Germany, May 1999. Springer-Verlag, Heidelberg.
5. Peter Haase, Andreas Hotho, Lars Schmidt-Thieme, and York Sure. Collaborative and usage-driven evolution of personal ontologies. In *Proc. of European Conference on the Semantic Web (ESWC 2005)*, LNCS 3532, pages 486–499. Springer Verlag Berlin Heidelberg, May/June 2005.
6. Siegfried Handschuh, Steffen Staab, and F. Ciravegna. S-CREAM – Semi-automatic CREation of metadata. In *Proc. of the European Conf. on Knowledge Acquisition and Management*, 2002.
7. Clyde Holsapple and K.D. Joshi. A collaborative approach to ontology design. *Communications of ACM*, 45(2):42–47, 2005.
8. Andreas Hotho, Steffen Staab, and Gerd Stumme. Explaining text clustering results using semantic structures. In *Proc. of ECML/PKDD 2003*, LNAI 2838, pages 217–228, Cavtat-Dubrovnik, Croatia, Sept. 2003. Springer Verlag.
9. M. Kavalec and V. Svatek. A study on automated relation labelling in ontology learning. In P. Buitelaar, P. Cimiano, and B. Magnini, editors, *Ontology Learning and Population*. IOS Press, 2005.
10. Jianming Li, Zhang Lei, and Yong Yu. Learning to generate semantic annotation for domain specific sentences. In *Proc. of the "Knowledge Markup and Semantic Annotation" Workshop of the K-CAP 2001 Conference*, 2001.
11. Alexander Maedche and Steffen Staab. Semi-automatic engineering of ontologies from text. In *Proc. of 12th Int. Conf. on Software and Knowledge Engineering*, Chicago, IL, 2000.
12. Roberto Navigli, Paola Velardi, Alessandro Cucchiarelli, and Francesca Neri. Quantitative and qualitative evaluation of the ontolearn ontology learning system. In *Proc. of ECAI-2004 Workshop on Ontology Learning and Population*, Valencia, Spain, Aug. 2004. http://olp.dfki.de/ecai04/cfp.htm, accessed at July 26, 2005.
13. Robert Porzel and Rainer Malaka. A task-based approach for ontology evaluation. In *Proc. of ECAI-2004 Workshop on Ontology Learning and Population*, Valencia, Spain, Aug. 2004. http://olp.dfki.de/ecai04/cfp.htm, accessed at July 26, 2005.
14. Markus Schaal, Roland Mueller, Marko Brunzel, and Myra Spiliopoulou. RELFIN - topic discovery for ontology enhancement and annotation. In *Proc. of European Conference on the Semantic Web (ESWC 2005)*, LNCS 3532, pages 608–622, Heraklion, Greece, May/June 2005. Springer Verlag Berlin Heidelberg.

15. Benno Stein, Sven Meyer zu Eissen, and Frank Wißbrock. On Cluster Validity and the Information Need of Users. In M.H. Hanza, editor, *3rd IASTED Int. Conference on Artificial Intelligence and Applications (AIA03)*, pages 216–221, Benalmadena, Spain, September 2003. ACTA Press.
16. Michalis Vazirgiannis, Maria Halkidi, and Dimitrios Gunopoulos. *Uncertainty Handling and Quality Assessment in Data Mining*. Springer, 2003.

Introducing Semantics in Web Personalization: The Role of Ontologies

Magdalini Eirinaki, Dimitrios Mavroeidis, George Tsatsaronis, and Michalis Vazirgiannis

Athens University of Economics and Business, Dept. of Informatics, Athens, Greece
{eirinaki, dmavr, gbt, mvazirg}@aueb.gr

Abstract. Web personalization is the process of customizing a web site to the needs of each specific user or set of users. Personalization of a web site may be performed by the provision of recommendations to the users, highlighting/adding links, creation of index pages, etc. The web personalization systems are mainly based on the exploitation of the navigational patterns of the web site's visitors. When a personalization system relies solely on usage-based results, however, valuable information conceptually related to what is finally recommended may be missed. The exploitation of the web pages' semantics can considerably improve the results of web usage mining and personalization, since it provides a more abstract yet uniform and both machine and human understandable way of processing and analyzing the usage data. The underlying idea is to integrate usage data with content semantics, expressed in ontology terms, in order to produce semantically enhanced navigational patterns that can subsequently be used for producing valuable recommendations. In this paper we propose a semantic web personalization system, focusing on word sense disambiguation techniques which can be applied in order to semantically annotate the web site's content.

1 Introduction

During the past few years the World Wide Web has emerged to become the biggest and most popular way of communication and information dissemination. Every day, the Web grows by roughly a million electronic pages, adding to the hundreds of millions pages already on-line. WWW serves as a platform for exchanging various kinds of information, ranging from research papers, and educational content, to multimedia content, software and personal weblogs (blogs). Because of its rapid and chaotic growth, the resulting network of information lacks of organization and structure. Users often feel disoriented and get lost in that information overload that continues to expand. On the other hand, the e-business sector is rapidly evolving and the need for Web market places that anticipate the needs of the customers is more than ever evident. Therefore, an ultimate need nowadays is that of predicting the user needs in order to improve the usability and user retention of a Web site.

In brief, web personalization can be defined as any action that adapts the information or services provided by a web site to an individual user, or a set of users, based

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 147 – 162, 2006.

on knowledge acquired by their *navigational behavior*, recorded in the web site's logs. This information is often combined with the *content* and the *structure* of the web site as well as the *user's interests/preferences*, if they are available. Using the four aforementioned sources of information as input to pattern discovery techniques, the system tailors the provided content to the needs of each visitor of the web site. The personalization process can result in the dynamic generation of recommendations, the creation of index pages, the highlighting of existing hyperlinks, the publishing of targeted advertisements or emails, etc. In this paper we focus on personalization systems that aim at providing personalized recommendations to the web site's visitors.

The problem of providing recommendations to the visitors of a web site has received a significant amount of attention in the related literature. Most of the earlier research efforts in Web personalization correspond to the evolution of extensive research in Web usage mining [3, 9, 41]. Pure usage-based personalization, however, presents certain shortcomings, for instance when there is not enough usage data available in order to extract patterns related to certain navigational actions, or when the web site's content changes and new pages are added but are not yet included in the web logs.

Motivated by the fact that the users' navigation is extremely semantically-driven, in other words the users' visits usually aim at finding information concerning a particular subject, we claim that the underlying content semantics should be a dominant factor in the process of web personalization. There have been a number of research studies that integrate the web site's content in order to enhance the web personalization process [18, 22, 30, 37]. Most of these efforts characterize web content by extracting features from the web pages. Usually these features are keywords subsequently used to retrieve similarly characterized content. The similarity between documents is usually based on exact matching between these terms. In this way, however, only a binary matching between documents is achieved, whereas no actual *semantic* similarity is taken into consideration.

The need for a more abstract representation that will enable a uniform and more flexible document matching process imposes the use of semantic web structures, such as ontologies[1] [6, 19]. By mapping the keywords to the concepts of an ontology, or topic hierarchy, the problem of binary matching can be surpassed through the use of the hierarchical relationships and/or the *semantic similarities* among the ontology terms, and therefore, the documents.

Several research studies proposed frameworks that express the users' navigational behavior in terms of an ontology and integrate this knowledge in semantic web sites [36], Markov model-based recommendation systems [2], or collaborative filtering systems [11, 33]. Overall, all the aforementioned approaches are based on the same intuition: enhance the web personalization process with content semantics, expressed using the terms of a domain-ontology. The extracted web content features are mapped to ontology terms and this abstraction enables the generalizations/specializations of the derived patterns and/or user profiles. In all proposed models, however, the ontology-term mapping process is performed manually or semi-automatically (needing the manual labeling of the training data set). Some approaches are based on collaborative filtering systems, which assume that some kind of user ratings are available, or on

[1] In this work we focus on the hierarchical part of an ontology. Therefore, in the rest of this work we use the terms *concept hierarchy, taxonomy* and *ontology* interchangeably.

semantic web sites, which assume that an existing underlying semantic annotation of the web content is available a priori. Finally, none of the aforementioned approaches fully exploits the underlying semantic similarities of terms belonging to an ontology, apart from the straightforward "is-a" or "parent-child" hierarchical relationships.

Since ontologies resemble the semantic networks underlying the word thesauri, the process of keyword mapping to ontology concepts can be related to thesaurus-based Word Sense Disambiguation (WSD). The analogy stems from the fact that both thesauri and ontologies contain a vast amount of semantic background information concerning the concepts they contain. The semantic information is usually expressed through semantic relations, such as "is-a" and "has-part" relations. Thesaurus-based WSD algorithms aim at exploiting such semantic relations for successfully mapping words to thesaurus concepts. Although the effectiveness of such methods for the semantic representation of documents had been an issue of controversy, recent thesaurus-based WSD algorithms have been shown to consistently improve the performance of classification and clustering tasks [7, 20, 29, 44].

In this paper we present a Semantic Web Personalization framework (further referred to as SEWeP) that integrates usage data with content semantics expressed in ontology terms in order to effectively generate useful recommendations. This framework is mainly based on the work presented in [14, 15, 38]. Similar to previously proposed approaches, the proposed personalization framework uses ontology terms to annotate the web content and the users' navigational patterns. The key departure from earlier approaches, however, is that the proposed personalization framework employs fully automatic ontology mapping WSD-based techniques [19, 29], by exploiting the underlying semantic similarities between ontology terms.

In the Section that follows we present work related to thesaurus-based WSD algorithms and web personalization systems which use ontologies. We then discuss several measures for computing similarity between ontology terms in Section 3. In Section 4 we present in detail the proposed Semantic Web Personalization framework and we conclude in Section 5.

2 Related Work

In this Section we present a short review on thesaurus-based WSD algorithms. We also review the research studies which integrate content data in the web personalization process, focusing on those that employ ontologies in order to represent the web documents.

2.1 Word Sense Disambiguation for Ontologies

In this subsection we present a short review on WSD approaches that are based on utilizing semantic relations in a word thesauri. Since ontologies resemble the semantic networks underlying a word thesaurus, these methods can be naturally extended for mapping keywords to ontology concepts. In the subsequent paragraph, we use WSD terminology with regards to a given word *w*. The "sense" of a word *w* is the concept of the thesaurus assigned to *w*. The "context" of *w*, refers to its surrounding words in the text it belongs to, and depending on the method its definition can vary and may

include from a small window of surrounding words, like in the method of Lesk [27], to all the words occurring in the same text as *w*, like in the method proposed in [34].

Several WSD approaches take advantage of the fact that a thesaurus offers important vertical (is-a, has-part) and horizontal (synonym, antonym, coordinate terms) semantic relations. Sussna [43] has proposed an unsupervised WSD approach where the distance among the candidate senses of a noun, as well as the senses of the words in its context are taken into account, using a sliding window of noun words occurring in the text. The correct sense that disambiguates each noun is found through minimizing the distance between possible assignments of senses of neighboring nouns. In order to compute this distance, the author considers a semantic distance measure which utilizes the semantic relations expressing the hypernym/hyponym, meronym/holonym and synonym/antonym nature. In the work of Agirre and Rigau [1] the hypernym/hyponym relation is used again to form a measure of conceptual distance between senses, by measuring the shortest path between the possible senses of a noun to be disambiguated and the senses of its context words. Rigau et al. [40] combined previous WSD approaches and utilized the hypernym/hyponym and the domain semantic relation, along with other heuristics that make use of measuring word co-occurrence in senses' definitions and constructing semantic vectors, to form a new unsupervised WSD approach. Leacock et. al. [25] have also used the hypernym/hyponym, the synonym and the coordinate terms semantic relationship (the latter expresses senses sharing the same immediate hypernym) existing in WordNet [16] to form the training material of their WSD algorithm. Mihalcea et al. [32] have used synonyms and hypernyms/hyponyms as well to generate the semantically connected senses for the words to be disambiguated. Their disambiguation takes place in an iterative manner, generating a set for the already disambiguated words and a set for ambiguous word, while utilizing possible semantic connections between the two sets. Montoyo et al. [31] also use hypernyms and hyponyms, along with their glosses, in order to combine knowledge-based and corpus-based WSD methods. Moreover, in [5, 17, 42] lexical chaining is used for word sense disambiguation, which is a process of connecting semantically related words (thus making use of hypernym/hyponym and other semantic relations) in order to create a set of chains that represent different threads of cohesion through a given text. Lexical chaining has also been validated in the area of text summarization [5, 42]. Finally, in [35], they use a variety of knowledge sources to automatically generate semantic graphs, which are essentially alternative conceptualizations for the lexical items to be disambiguated. For building these graphs they used various types of semantic relations, like meronymy/holonymy, hypernymy/hyponymy and synonymy.

In contrast to the approaches described above, the WSD algorithm proposed in [29] has been validated experimentally both in "pure" WSD (using WSD benchmark datasets), and in the document classification task. The fact that our approach has been shown to improve classification accuracy, constitutes a strong indication that it can be used effectively for enhancing semantics in document representation.

2.2 Using Content Semantics for Web Personalization

Several frameworks based on the claim that the incorporation of information related to the web site's content enhances the web mining and web personalization process

have been proposed prior [30, 37] or subsequent [18, 22, 23] to our original work [14, 15]. In this subsection we overview in detail the ones that are more similar to ours, in terms of using a domain-ontology to represent the web site's content for enhancing the web personalization results.

Dai and Mobasher [11] proposed a web personalization framework that characterizes the usage profiles of a collaborative filtering system using ontologies. These profiles are transformed to "domain-level" aggregate profiles by representing each page with a set of related ontology objects. In this work, the mapping of content features to ontology terms is assumed to be performed either manually, or using supervised learning methods. The defined ontology includes classes and their instances therefore the aggregation is performed by grouping together different instances that belong to the same class. The recommendations generated by the proposed collaborative system are in turn derived by binary matching the current user visit expressed as ontology instances to the derived domain-level aggregate profiles, and no semantic relations beyond hyperonymy/hyponymy are employed.

The idea of semantically enhancing the web logs using ontology concepts is independently described by Oberle et.al. [36]. This framework is based on a semantic web site built on an underlying ontology. This site contains both static and dynamic pages being generated out of the ontology. The authors present a general framework where data mining can then be performed on these semantic web logs to extract knowledge about groups of users, users' preferences, and rules. Since the proposed framework is built on a semantic web knowledge portal, the web content is inherently semantically-annotated exploiting the portal's inherent RDF annotations. The authors discuss how this framework can be extended using generalizations/specializations of the ontology terms, as well as for supporting the web personalization process, yet they mainly focus on web mining.

Acharyya and Ghosh [2] also propose a general personalization framework based on the conceptual modeling of the users' navigational behavior. The proposed methodology involves mapping each visited page to a topic or concept, imposing a tree hierarchy (taxonomy) on these topics, and then estimating the parameters of a semi-Markov process defined on this tree based on the observed user paths. In this Markov models-based work, the semantic characterization of the context is performed manually. Moreover, no semantic similarity measure is exploited for enhancing the prediction process, except for generalizations/specializations of the ontology terms.

Middleton et. al [33] explore the use of ontologies in the user profiling process within collaborative filtering systems. This work focuses on recommending academic research papers to academic staff of a University. The authors represent the acquired user profiles using terms of a research paper ontology (is-a hierarchy). Research papers are also classified using ontological classes. In this hybrid recommender system which is based on collaborative and content-based recommendation techniques, the content is characterized with ontology terms, using document classifiers (therefore a manual labeling of the training set is needed) and the ontology is again used for making generalizations/specializations of the user profiles.

Finally, Kearney and Anand [23] use an ontology to calculate the impact of different ontology concepts on the users navigational behavior (selection of items). In this work, they suggest that these impact values can be used to more accurately determine distance between different users as well as between user preferences and other items

on the web site, two basic operations carried out in content and collaborative filtering based recommendations. The similarity measure they employ is very similar to the Wu & Palmer similarity measure presented here. This work focuses on the way these ontological profiles are created, rather than evaluating their impact in the recommendation process, which remains opens for future work.

3 Similarity of Ontology Terms

As already mentioned, the proposed semantic web personalization framework exploits the expressive power of content semantics, that are represented by ontology terms. Using such a representation, the similarity between documents is deduced to the distance between terms that are part of a hierarchy. The need for such a similarity measure is encountered throughout the personalization process, namely during content characterization, keyword translation, document clustering and recommendations' generation.

There is an extensive bibliography addressing the issue of defining semantic distances and similarity measures based on semantic relations. A popular similarity measure for ontology concepts is proposed by Resnik [39]. The similarity between two ontology concepts is based on the "depth" of their least common ancestor, where the "depth" is measured using the information content. Formally the similarity measure is defined as: $RSsim(a,b) = \max_{\{c \in Supp(a,b)\}} IC(c)$, where $IC(c) = -\log P(c)$ is the information content of concept *c* and *Supp(a,b)* is a set containing all ancestors (in the hierarchical structure) of *a* and *b*.

Jiang and Conrath [21] define a distance measure based on the path of two concepts to their least common ancestor. Their distance measure does not depend solely on the edge counting, since the information content is used for weighting the edges. Formally the Jiang and Conrath distance measure is defined as: $JCdis(a,b) = IC(a) + IC(b) - 2IC(lca(a,b))$, where *IC(c)* is the information content of concept *c* and *lca(a,b)* is the least common ancestor of *a* and *b*.

Leacock and Chodorow [24] define a similarity measure that is based on the shortest path that connects two concepts normalized by the maximum depth of the ontology. Their similarity measure is defined as: $LCsim(a,b) = -\log \frac{path\_length(a,b)}{2D}$, where *path_length* is the length of the shortest path that connects the two concepts in the ontology and *D* denotes the maximum depth of the ontology.

Finally, Lin [28] has proposed a similarity measure, based on the Wu and Palmer similarity measure [48]. More precisely, they incorporated the information content in order to extend the flexibility of the similarity measure beyond edge counting, The Lin similarity measure is defined as: $LinSim(a,b) = \frac{2IC(lca(a,b))}{IC(a) + IC(b)}$, where *lca(a,b)* defines the least common ancestor of concepts *a* and *b*.

The ontology similarity and distance measures described above are defined for pairs of concepts that reside in an ontology and cannot be directly used for evaluating the similarity between documents (a document contains a set of concepts). However, they can be used for evaluating the similarity between two documents either in the

context of distance measures for sets of objects, or in the context of the Generalized Vector Space Model (GVSM model) [29, 37].

In our approach, we adopt the Wu & Palmer similarity measure [48] for calculating the distance between terms that belong to a tree (hierarchy). Moreover, we use its generalization, proposed by Halkidi et.al. [19] to compute the similarity between sets of terms that belong to a concept hierarchy. Furthermore we utilize a recently proposed similarity measure for sets of ontology concepts that is based on the GVSM model proposed in [29]. We should stress that the choice of the similarity measure is orthogonal to the rest system functionality, as long as it serves for calculating the distance between hierarchically organized terms. The definitions of the three similarity measures are given in what follows. A more detailed description and theoretical proof can be found in the related publications.

3.1 Wu&Palmer Similarity Measure

Given a tree, and two nodes *a, b* of this tree, their similarity is computed as follows:

$$WPsim(a,b) = \frac{depth(a) + depth(b)}{2 * depth(c)} \quad (1)$$

where the node *c* is their deepest (in terms of tree depth) common ancestor.

3.2 THESUS Similarity Measure

Given an ontology *T* and two sets of weighted terms $\mathcal{A}=\{(w_i, k_i)\}$ and $\mathcal{B}=\{(v_i, h_i)\}$, with $w_i, v_i \in T$, their similarity is defined as:

$$THEsim(\mathcal{A},\mathcal{B}) = \frac{1}{2}\left[\left(\frac{1}{K}\sum_{i=1}^{|\mathcal{A}|}\max_{j\in[1,|\mathcal{B}|]}\left(\lambda_{i,j}\times WPsim(k_i,h_j)\right)\right)+\left(\frac{1}{H}\sum_{i=1}^{|\mathcal{B}|}\max_{j\in[1,|\mathcal{A}|]}\left(\mu_{i,j}\times WPsim(h_i,k_j)\right)\right)\right] \quad (2)$$

where $\lambda_{i,j} = \frac{w_i + v_j}{2\times\max(w_i, v_j)}$ and $K = \sum_{i=1}^{|A|}\lambda_{i,x(i)}$,with

$x(i) = x \mid \lambda_{i,x}\times WPsim(k_i,h_x) = \max_{j\in[1,|B|]}\left(\lambda_{i,x}\times WPsim(k_i,h_j)\right)$.

The theoretical and experimental justification of the effectiveness of the aforementioned similarity measure is included in [19].

3.3 GVSM Based Similarity Measure

The similarity between two documents $d_1$and d_2 that contain ontology concepts is defined as

$$GVSMsim(d_1 d_2) = d_1 DD^T d_2^T \quad (3)$$

where the rows of matrix *D* contain the vector representations of the ontology concepts. For constructing the vector representations, we initially consider an index of all the ontology concepts. Then the vector representation of each concept has non-zero elements only at the dimensions that correspond to the concept's ancestors. For illustrative purposes, consider two ontology concepts c_1=*insurance_company* and

c_2=*bank*, where both concepts have two ancestors in the ontology hierarchy, c_3=*financial_insitution* and c_4=*institution*. Then, if we consider that the concept hierarchy contains only these four concepts, the index of all the ontology concepts will be (c_1, c_2, c_3, c_4), and the vector representation of c_1 will be *(1,0,1,1)*, while the vector representation of c_2 will be *(0,1,1,1)*. In [29] we justify theoretically (Propositions 1, 2) and experimentally the effectiveness of the proposed similarity measure.

4 Ontology-Based Semantic Web Personalization

The users' navigation in a web site is usually content-oriented. The users often search for information or services concerning a particular topic. Therefore, the underlying content semantics should be a dominant factor in the process of web personalization. In this Section we present a web personalization framework that integrates content semantics with the users' navigational patterns, using ontologies to represent both the content and the usage of the web site. This framework is mainly based on the SEWeP personalization system, presented in [14, 15, 38]. To the best of our knowledge, it is the only web personalization framework where the content characterization process is performed using WSD-based methods [19, 29], fully exploiting the underlying semantic similarities of ontology terms.

4.1 SEWeP System Architecture

SEWeP uses a combination of web mining techniques to personalize a web site. In short, the web site's content is processed and characterized by a set of ontology terms (categories). The visitors' navigational behavior is also updated with this semantic knowledge to create an enhanced version of web logs, C-logs, as well as semantic document clusters. C-Logs are in turn mined to produce both a set of URI and category-based association rules. Finally, the recommendation engine uses these rules, along with the semantic document clusters in order to provide the final, semantically enhanced set of recommendations to the end user.

As illustrated in Figure 1, SEWeP consists of the following components:

- *Content Characterization.* This module takes as input the content of the web site as well as a domain-specific ontology and outputs the semantically annotated content to the modules that are responsible for creating the C-Logs and the semantic document clusters.
- *Semantic Document Clustering.* The semantically annotated pages created by the previous component are grouped into thematic clusters. This categorization is achieved by clustering the web documents based on the semantic similarity between the ontology terms that characterize them.
- *C-Logs Creation & Mining.* This module takes as input the web site's logs as well as the semantically annotated web site content. It outputs both URI and category-based frequent itemsets and association rules which are subsequently matched to the current user's visit by the recommendation engine.
- *Recommendation Engine.* This module takes as input the current user's path and matches it with the semantically annotated navigational patterns produced in

the previous phases. The recommendation engine generates three different recommendation sets, namely *original*, *semantic* and *category-based* ones, depending on the input patterns used.

The creation of the ontology as well as the semantic similarity measures used as input in the aforementioned web personalization process are orthogonal to the proposed framework. We assume that the ontology is descriptive of the web site's domain and is provided/created by a domain expert. We elaborated on several similarity measures for ontology terms in Section 3. In what follows we briefly describe the key components of the proposed architecture. For more details on the respective algorithms and system implementation the reader may refer to [14, 15, 38].

4.2 Content Characterization

A fundamental component of the SEWeP architecture is the automatic content characterization process. SEWeP is the only web personalization framework enabling the automatic annotation of web content with ontology terms without needing any human labeling or prior training of the system. The keywords' extraction is based both on the content of the web pages, as well as their connectivity features. What is more, SEWeP enables the annotation of multilingual content, since it incorporates a context-sensitive translation component which can be applied prior to the ontology mapping process. In the subsections that follow we briefly describe the aforementioned processes, namely the keyword extraction and translation as well as the semantic characterization modules.

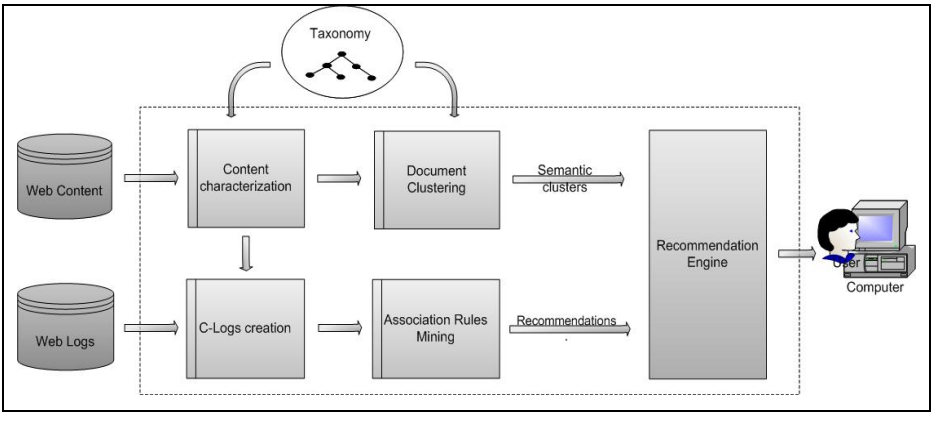


Fig. 1. SEWeP architecture

4.2.1 Keyword Extraction

There exists a wealth of methods for representing web documents. The most straightforward approach is to perform text mining in the document itself following standard IR techniques. This approach, however, proves insufficient for the web content, since it relies solely on the information included in the document ignoring semantics arising from the connectivity features of the web [8, 10]. Therefore, in many approaches

information contained in the links that point to the document and the text near them - defined as "anchor-window" - is used for characterizing a web document [10, 46, 45]. This approach is based on the hypothesis that the text around the link to a page is descriptive of the page's contents and overcomes the problems of the content-based approach, since it takes into consideration the way others characterize a specific web page. In our work, we adopt and extend this approach, by also taking into consideration the content of the pages that are pointed by the page under examination, based on the assumption that in most Web pages the authors include links to topics that are of importance in the page's context.

In the proposed framework, the keywords that characterize a web page *p* are extracted using:

1. The raw term frequency of *p*.
2. The raw term frequency of a selected fraction (anchor-window) of the web pages that point to *p*.
3. The raw term frequency of the web pages that are pointed by *p*.

This hybrid content & structure –based keyword extraction process is motivated by the fact that the text around the links pointing to the web page, as well as the content of the web pages pointed by the web page under consideration are descriptive of the page's contents.

At the end of this phase, each document *d* is characterized by a weighted set of keywords $d = \{(k_i,w_i)\}$, where w_i is the weight representing the summed (over the combination of methods) word frequency of keyword k_i. Before proceeding with mapping the extracted keywords to related ontology terms, however, all non-English keywords should be translated. In our approach, we determine the most suitable synonym using a context-sensitive automatic translation method. Assuming that the set of keywords will be descriptive of the web page's content, we derive the best synonym set by comparing their semantics. This translation method is applicable for any language, provided that a dictionary and its inflection rules are available. In our system implementation we applied it for the Greek language. More details on this algorithm can be found in [14, 26].

4.2.2 Semantic Characterization

In order to assist the remainder of the personalization process (C-logs creation, semantic document clustering, semantic recommendations) the *n* most frequent (translated) keywords that where extracted in the previous phase, are mapped to the terms $T = \{c_1, \ldots, c_k\}$ of a domain ontology (in our approach we need the concept hierarchy part of the ontology). This mapping is performed using a thesaurus, like Wordnet [16]. If the keyword belongs to the ontology, then it is included as it is. Otherwise, the system finds the "closest" (i.e. most similar) term (*category*) to the keyword using the unsupervised WSD algorithm proposed in [29]. This algorithm adopts the intuition that context terms (adjacent term in text) are semantically close to each other and that this is reflected by their pathwise distance on the hierarchical structure of the ontology. Based on this intuition the WSD algorithm maps a set of terms to the ontology concepts that minimize the pathwize distances on the ontology hierarchical structure. Thus, the objective of our WSD algorithm is to find the set of senses (among the candidate sets of senses) that is more "compact" in the ontology structure. The compactness measure utilized for selecting the

appropriate set of senses is based on the concept of the Steiner Tree (minimum-weight Tree that connects a set of vertices in a Graph).

If more than one keywords are mapped to the same category c_i, the relevance r_i assigned to it is computed using the following formula:

$$r_i = \sum_{k_j \to c_i} (w_j \cdot s_j) \Big/ \sum_{k_j \to c_i} w_j \tag{4}$$

where w_j is the weight assigned to keyword k_j for document d and s_j the similarity with which k_j is mapped to c_i. At the end of this process, each document d is represented as a set $d = \{(c_i, r_i)\}$, where $r_i \in [0,1]$ since $s_j \in [0,1]$. If only one keyword k_j is mapped to a category c_i, then the respective relevance r_i equals the keyword's weight w_j.

4.3 C-Logs Creation and Mining

C-Logs are in essence an enhanced form of the web logs. The C-Logs creation process involves the correlation of each web logs' record with the ontology terms that represent the respective URI. C-logs may be further processed in the same way as web logs, through the use of statistical and data mining techniques, such as association rules, clustering or sequential pattern discovery.

The web mining algorithms currently supported by SEWeP is frequent itemsets' and association rules' discovery. Both algorithms are based on a variation of the Apriori algorithm [4], used to extract patterns that represent the visitors' navigational behavior in terms of pages often visited together. The input to the algorithm is the recorded users' sessions expressed both in URI and category level. The output is a set of URI and category-based frequent itemsets or association rules respectively. Since no explicit user/session identification data are available, we assume that a session is defined by all the pageview visits made by the same IP, having less than a maximum threshold time gap between consecutive hits.

4.4 Document Clustering

After the content characterization process, all web documents are semantically annotated with terms belonging to a concept hierarchy. This knowledge is materialized by grouping together documents that are characterized by semantically "close" terms, i.e. neighboring categories in the hierarchy. This categorization is achieved by clustering the web documents based on the similarity among the ontology terms that characterize each one of them. The generated clusters capture semantic relationships that may not be obvious at first sight, for example documents that are not "structurally" close (i.e. under the same root path).

For this purpose we use the THESUS similarity measure, as defined earlier, with a modification of the density-based algorithm DBSCAN [13] for clustering the documents. After the document clustering, each cluster is labeled by the most descriptive categories of the documents it contains, i.e. the categories that characterize more than *t%* of the documents. Modification details and the algorithm itself are described in [19, 46]. The semantic document clusters are used in turn in order to expand the recommendation set with semantically similar web pages, as we describe in the subsequent Section.

4.5 Recommendation Engine

As already mentioned, after the document characterization and clustering processes have been completed, each document d is represented by a set of weighted terms (categories) that are part of the concept hierarchy: $d = \{(c_i, r_i)\}$, $c_i \in T$, $r_i \in [0,1]$ (T is the concept hierarchy, r_i is c_i's weight). This knowledge can be transformed into three different types of recommendations, depending on the rules that are used as input (association rules between URIs or between categories) and the involvement of semantic document clusters: *original*, *semantic*, and *category-based recommendations.*

Original recommendations are the "straightforward" way of producing recommendations, simply relying in the usage data of a web site. They are produced when, for each incoming user, a sliding window of her past *n* visits is matched to the *URI-based* association rules in the database, and the *m* most similar ones are selected. The system recommends the URIs included in the rules, but not visited by the user so far.

The intuition behind *semantic recommendations* is that, useful knowledge semantically similar to the one originally proposed to the users, is omitted for several reasons (updated content, not enough usage data etc.) Those recommendations are in the same format as the *original* ones but the web personalization process is enhanced by taking into account the semantic proximity of the content. In this way, the system's suggestions are enriched with content bearing similar semantics. In short, they are produced when, for each incoming user, a sliding window of her past *n* visits is matched to the *URI-based* association rules in the database, and the single most similar one is selected. The system finds the URIs included in the rule but not yet visited by the user (let A) and recommends the *m* most similar documents that are in the same semantic cluster as A.

Finally, the intuition behind *category-based* recommendations is the same as the one of *semantic* recommendations: incorporate content and usage data in the recommendation process. This notion, however, is further expanded; users' navigational behavior is now expressed using a more abstract, yet semantically meaningful way. Both the navigational patterns' knowledge database and the current user's profile are expressed by categories. Therefore, pattern matching to the current user's navigational behavior is no longer exact since it utilizes the semantic relationships between the categories, as expressed by their topology in the domain-specific concept hierarchy. The final set of recommendations is produced when, for each incoming user, a sliding window of the user's past *n* visits is matched to the category-based association rules in the database, and the most similar is selected. The system finds the most relevant document cluster (using similarity between category terms) and recommends the documents that are not yet visited by the user.

In what follows, we briefly describe the *semantic* and *category-based recommendations'* algorithms. The description of the generation of original recommendations is omitted, since it is a straightforward application of the Apriori [4] algorithm to the sessionized web logs. The respective algorithms, as well as experimental evaluation of the proposed framework can be found in [12, 14, 15].

4.5.1 Semantic Recommendations

We use the Apriori algorithm to discover frequent itemsets and/or association rules from the C-Logs. We consider that each distinct user session represents a different

transaction. We will use $S = \{I_m\}$, to denote the final set of frequent itemsets/association rules, where $I_m = \{(uri_i)\}, uri_i \in CL$.

The recommendation method takes as input the user's current visit, expressed a set of URIs: $CV = \{(uri_j)\}, uri_j \in WS$, (*WS* is the set of URIs in the visited web site. Note that some of these may not be included in *CL*). The method finds the itemset in *S* that is most similar to *CV*, and recommends the documents (labeled by related categories) belonging to the most similar document cluster $Cl_m \in Cl$ (*Cl* is the set of document clusters). In order to find the similarity between URIs, we perform binary matching. In other words, the more common URIs in *CV* and *S*, the more similar they are.

4.5.2 Category-Based Recommendations

We use an adaptation of the Apriori algorithm to discover frequent itemsets and/or association rules including categories. We consider that each distinct user session represents a different transaction. Instead of using as input the distinct URIs visited, we replace them with the respective categories. We keep the most important ones, based on their frequency (since the same category may characterize more than one documents). We then apply the Apriori algorithm using categories as items. We will use $C = \{ I_k \}$, to denote the final set of frequent itemsets/association rules, where $I_k = \{(c_{i,}, r_i)\}, r_i \in T, r_i \in [0,1]$ (r_i reflects the frequency of c_i).

The recommendation method takes as input the user's current visit, expressed in weighted category terms: $CV = \{(c_j, f_j)\}, c_j \in T, f_j \in [0,1]$ (f_j is frequency of c_j in current visit - normalized). The method finds the itemset in *C* that is most similar to *CV*, creates a generalization of it and recommends the documents (labeled by related categories) belonging to the most similar document cluster $Cl_n \in Cl$ (*Cl* is the set of document clusters). To find the similarity between categories we use the Wu & Palmer metric, whereas in order to find similarity between sets of categories, we use the THESUS metric, as defined in Section 3.

5 Conclusions

The exploitation of the pages' semantics hidden in user paths can considerably improve the results of web personalization, since it provides a more abstract yet uniform and both machine and human understandable way of processing and analyzing the data. In this paper, we present a semantic web personalization framework, which enhances the recommendation process with content semantics. We focus on word sense disambiguation techniques which can be used in order to semantically annotate the web site's content with ontology terms. The framework exploits the inherent semantic similarities between the ontology terms in order to group web documents together and semantically expand the recommendation set.

A more detailed analysis, possible limitations and an extensive experimental evaluation of the several components of the proposed framework can be found in [14, 15, 29]. The experimental results are more than promising. Our plans for future work include the evaluation of the proposed integrated SEWeP-GVSM framework. We also plan to integrate different semantic similarity measures in our architecture.

References

1. E. Agirre, G. Rigau, *A proposal for word sense disambiguation using conceptual distance*, In Proc. of Recent Advances in NLP (RANLP), 1995, pp. 258–264
2. S. Acharyya, J. Ghosh, *Context-Sensitive Modeling of Web Surfing Behaviour Using Concept Trees*, in Proc. of the 5th WEBKDD Workshop, Washington, August 2003
3. M. Albanese, A. Picariello, C. Sansone, L. Sansone, *A Web Personalization System based on Web Usage Mining Techniques*, in Proc. of WWW2004, May 2004, New York, USA
4. R. Agrawal, R. Srikant, *Fast Algorithms for Mining Association Rules*, in Proc. of 20th VLDB Conference, 1994
5. R. Barzilay, M. Elhadad, *Using lexical chains for text summarization*, in Proc. of the Intelligent Scalable Text Summarization Workshop (ISTS 1997), ACL, 1997.
6. B. Berendt, A. Hotho, G. Stumme, *Towards Semantic Web Mining*, in Proc. of 1st International Semantic Web Conference (ISWC 2002)
7. S. Bloehdorn, A. Hotho: *Boosting for text classification with semantic features*. In: Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Mining for and from the Semantic Web Workshop. (2004) 70–87
8. S. Brin, L. Page, *The anatomy of a large-scale hypertextual Web search engine*, Computer Networks, 30(1-7): 107-117, 1998, Proc. of the 7th International World Wide Web Conference (WWW7)
9. R. Baraglia, F. Silvestri, *An Online Recommender System for Large Web Sites*, in Proc. of ACM/IEEE Web Intelligence Conference (WI'04), China, September 2004
10. S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, J. Kleinberg, *Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text*, in Proc. of WWW7, 1998
11. H. Dai, B. Mobasher, *Using Ontologies to Discover Domain-Level Web Usage Profiles*, in Proc. of the 2nd Workshop on Semantic Web Mining, Helsinki, Finland, 2002
12. M. Eirinaki, *New Approaches to Web Personalization*, PhD Thesis, Athens University of Economics and Business, Dept. of Informatics, 2006
13. M. Ester, H.P. Kriegel, J. Sander, M. Wimmer and X. Xu, *Incremental Clustering for Mining in a Data Warehousing Environment*, in Proc. of the 24th VLDB Conference, 1998
14. M. Eirinaki, C. Lampos, S. Pavlakis, M. Vazirgiannis, *Web Personalization Integrating Content Semantics and Navigational Patterns*, in Proc. of the 6th ACM International Workshop on Web Information and Data Management (WIDM'04), November 2004, Washington DC
15. M. Eirinaki, M. Vazirgiannis, I. Varlamis, *SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process*, in Proc. of the 9th ACM International Conference on Knowledge Discovery and Data Mining (KDD '03), Washington DC, August 2003
16. C. Fellbaum, ed., *WordNet, An Electronic Lexical Database*. The MIT Press, 1998
17. M. Galley, K. McKeown, *Improving Word Sense Disambiguation in Lexical Chaining*, in Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), August 2003, Acapulco, Mexico.
18. J. Guo, V. Keselj, Q. Gao, *Integrating Web Content Clustering into Web Log Association Rule Mining*. In Proc. of Canadian AI Conference 2005
19. M. Halkidi, B. Nguyen, I. Varlamis, M. Vazirgiannis, *THESUS: Organizing Web Documents into Thematic Subsets using an Ontology*, VLDB journal, vol.12, No.4, 320-332, Nov. 2003

20. A. Hotho, S. Staab, G. Stumme: *Ontologies Improve Text Document Clustering*. Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA: 541-544
21. J. Jiang, D. Conrath, *Semantic similarity based on corpus statistics and lexical taxonomy*, in Proc. of the International Conference on Research in Computational Linguistics, 1997
22. X. Jin, Y. Zhou, B. Mobasher, *A Maximum Entropy Web Recommendation System: Combining Collaborative and Content Features*, in Proc. of the 11th ACM International Conference on Knowledge Discovery and Data Mining (KDD '05), Chicago, August 2005
23. P. Kearney, S. S. Anand, *Employing a Domain Ontology to gain insights into user behaviour*, in Proc. of the 3rd Workshop on Intelligent Techniques for Web Personalization (ITWP 2005), Endiburgh, Scotland, August 2005
24. C. Leacock, M. Chodorow. *Combining Local Context and WordNet Similarity for Word Sense Identification*. In Fellbaum 1998, pp. 265--283.
25. C. Leacock, M. Chodorow, G. A. Miller. 1998. *Using Corpus Statistics and WordNet Relations for Sense Identification*. In Computational Linguistics, 24:1 pp. 147–165.
26. C. Lampos, M. Eirinaki, D. Jevtuchova, M. Vazirgiannis, *Archiving the Greek Web*, in Proc. of the 4th International Web Archiving Workshop (IWAW04), September 2004, Bath, UK
27. M. E. Lesk, *Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone* , in Proc. of the SIGDOC Conference, June 1986, Toronto.
28. D. Lin, *An information-theoretic definition of similarity*, in Proc. of the 15th International Conference on Machine Learning (ICML), 1998, pp. 296–304
29. D. Mavroeidis, G. Tsatsaronis, M. Vazirgiannis, M. Theobald, G. Weikum, *Word Sense Disambiguation for Exploiting Hierarchical Thesauri in Text Classification*, in Proc. of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, (PKDD'05), Porto, Portugal, 2005
30. B. Mobasher, H. Dai, T. Luo, Y. Sung, J. Zhu, *Integrating web usage and content mining for more effective personalization*, in Proc. of the International Conference on Ecommerce and Web Technologies (ECWeb2000), Greenwich, UK, September 2000
31. A. Montoyo, A. Suarez, G. Rigau, M. Palomar, *Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods*, Journal of Artificial Intelligence Research, 23, pp.299–330.
32. R. Mihalcea, D. I. Moldovan, *A Highly Accurate Bootstrapping Algorithm for Word Sense Disambiguation*, International Journal on Artificial Intelligence Tools, 2001, 10(1-2), pp. 5–21.
33. S.E. Middleton, N.R. Shadbolt, D.C. De Roure, *Ontological User Profiling in Recommender Systems*, ACM Transactions on Information Systems (TOIS), Jan. 2004/ Vol.22, No. 1, 54-88
34. R. Mihalcea, P. Tarau, E. Figa, *Pagerank on semantic networks, with application to word sense disambiguation*, in Proc. of the 20th International Conference on Computational Linguistics (COLING 2004), August 2004, Switzerland.
35. R. Navigli, P. Velardi, *Structural Semantic Interconnection: a knowledge-based approach to Word Sense Disambiguation*, IEEE Transactions on Pattern Analysis and Machine Intelligence (TRAMI), 27(7), p. 1075-1086.
36. D.Oberle, B.Berendt, A.Hotho, J.Gonzalez, *Conceptual User Tracking*, in Proc. of the 1st Atlantic Web Intelligence Conf. (AWIC), 2003
37. M. Perkowitz, O. Etzioni, *Towards Adaptive Web Sites: Conceptual Framework and Case Study*, in Artificial Intelligence 118[1-2] (2000), pp. 245-275

38. S. Paulakis, C. Lampos, M. Eirinaki, M. Vazirgiannis, *SEWeP: A Web Mining System supporting Semantic Personalization*, in Proc. of the ECML/PKDD 2004 Conference, Pisa, Italy, September 2004
39. P. Resnik, *Using information content to evaluate semantic similarity in a taxonomy*, in Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI), 1995
40. G. Rigau, J. Atserias, E. Agirre. 1997.*Combining Unsupervised Lexical Knowledge Methods for Word Sense Disambiguation*, in Proc. of joint 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain.
41. B. Sarwar, G. Karypis, J. Konstan, J. Riedl, *Item-based Collaborative Filtering Recommendation Algorithms*, in Proc. of WWW10, May 2001, Hong Kong
42. G. Silber, K. McCoy, *Efficiently computed lexical chains as an intermediate representation for automatic text summarization*, Computational Linguistics, 29(1), 2003.
43. M. Sussna, *Word sense disambiguation for free-text indexing using a massive semantic network*. In: Proc. of the 2nd International Conference on Information and Knowledge Management (CIKM). (1993) 67–74
44. Theobald, M., Schenkel, R., Weikum, G.: *Exploiting structure, annotation, and ontological knowledge for automatic classification of xml data*. In: International Workshop on Web and Databases (WebDB). (2003) 1–6
45. H. Utard, J. Furnkranz, *Link-Local Features for Hypertext Classification*, in Proc. of the European Web Mining Forum (EWMF 2005), Porto, Portugal, October 2005
46. I. Varlamis, M. Vazirgiannis, M. Halkidi, B. Nguyen, *THESUS, A Closer View on Web Content Management Enhanced with Link Semantics*, in IEEE Transactions on Knowledge and Data Engineering Journal, June 2004 (Vol. 16, No. 6), pp. 585-600.
47. S.K.M. Wong, W. Ziarko, P.C.N. Wong, *Generalized vector space model in information retrieval*, in Proc. of the 8th Intl. ACM SIGIR Conference (SIGIR'85), 1985, pp. 18–25
48. Z. Wu, M. Palmer, *Verb Semantics and Lexical Selection*, 32nd Annual Meetings of the Associations for Computational Linguistics, 1994, pp. 133-138

Ontology-Enhanced Association Mining

Vojtěch Svátek[1], Jan Rauch[1], and Martin Ralbovský[2]

[1] Department of Information and Knowledge Engineering,
University of Economics, Prague, W. Churchill Sq. 4, 130 67 Praha 3, Czech Republic
svatek@vse.cz, rauch@vse.cz
[2] Faculty of Mathematics and Physics, Charles University in Prague, Ke Karlovu 3,
121 16 Praha 2, Czech Republic
martin.ralbovsky@gmail.com

Abstract. The roles of ontologies in KDD are potentially manifold. We track them through different phases of the KDD process, from data understanding through task setting to mining result interpretation and sharing over the semantic web. The underlying KDD paradigm is association mining tailored to our *4ft-Miner* tool. Experience from two different application domains—medicine and sociology—is presented throughout the paper. Envisaged software support for prior knowledge exploitation via customisation of an existing user-oriented KDD tool is also discussed.

1 Introduction

Domain ontologies, being hot topic in today's knowledge engineering research, are promising candidates for background knowledge to be used in the KDD process. They express the main concepts and relationships in a domain in a way that is consensual and comprehensible to the given professional community, and (ideally) commits to some generic principles of knowledge organisation. The research in applied ontology and in KDD are, to some extent, two sides of the same coin. Ontologies describe the 'state-of-affairs' in a certain domain at an abstract level, and thus enable to verify the correctness of existing (concrete) facts as well as to infer new facts. On the other hand, KDD typically proceeds in the opposite direction: from concrete, instance-level patterns to more abstract ones. Semantic web mining [5] represents the junction of ontology and KDD research in their 'concrete' (instance-centric) corners; in this paper, we however mostly focus on the junction of 'abstract' corners, namely, of abstract ontologies themselves and data generalisations (i.e. discovered hypotheses) produced by KDD.

The role to be played by ontologies in KDD (and even their mere usability) depends on the given mining *task* and *method*, on the *stage of the KDD process*, and also on some characteristics of the *domain* and *dataset*. The experiment described in this paper is connected with *task* of *association mining*, namely, with the *4ft-Miner* tool [23] (component of *LISp-Miner*, see http://lispminer.vse.cz), which is inspired by the GUHA *method* [13]. We identified four *stages* of (*4ft-Miner*-based) KDD that are likely to benefit from ontology application: data understanding, task design, result interpretation and result dissemination over

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 163–179, 2006.

the semantic web[1]. Finally, we conducted our research in two different *domains* with specific *datasets* (and available ontological resources): the domain of cardiovascular risk and that of social climate.

The paper is structured as follows. Section 2 describes both domain-specific applications. Section 3 recalls the basic principles of *4ft-Miner*. Sections 4, 5, 6 and 7 are devoted each to one phase of the KDD process as outlined above. Finally, section 9 reviews some related work, and section 10 shows directions for future research.

2 Overview of Applications

2.1 Cardiovascular Risk: Data and Ontologies

The *STULONG dataset* concerns a twenty-years-lasting longitudinal study of risk factors for atherosclerosis in the population of middle-aged men (see `http://euromise.vse.cz/stulong-en/`). It consists of four data matrices:

Entrance. Each of 1 417 men has been subject to entrance examination. Values of 244 attributes have been surveyed with each patient. These attributes are divided into 11 groups e. g. *social characteristics*, *physical activity* etc.

Control. Risk factors and clinical demonstration of atherosclerosis have been followed during the control examination for the duration of 20 years. Values of 66 attributes have been recorded for each one. There are 6 groups of attributes, e.g. *physical examination*, *biochemical examination* etc.

Letter. Additional information about health status of 403 men was collected by postal questionnaire. There are 62 attributes divided into 8 groups such as *diet* or *smoking*.

Death. There are 5 attributes concerning the death of 389 patients.

As ontology we used *UMLS* (Unified Medical Language System) [2], namely its high-level *semantic network* and the *meta-thesaurus* mapping the concepts picked from third-party resources onto each other. Although the central construct of UMLS is the concept-subconcept relation, the semantic network also features lots of other binary relations such as 'location of' or 'produces'. However, since the network only covers 134 high-level 'semantic types' (such as 'Body Part' or 'Disease'), the relations are only 'potentially holding' (it is by far not true that every Body Part can be 'location of' every Disease...). The meta-thesaurus, in turn, covers (a large number of) more specific concepts but relations are only scarcely instantiated, and nearly all relation instances belong to the 'location of' relation.

As additional resource, we used the knowledge accumulated in the Czech medical community with respect to risk factors of cardio-vascular diseases, in connection with the STULONG project itself. The knowledge base consists of

[1] In a pre-cursor paper [9], we explicitly used *CRISP-DM* (`http://www.crisp-dm.org`) for process decomposition; however, the phases are rather generic.

36 *qualitative rules*, most of which can be characterised as medical 'field knowledge' or common-sense knowledge, e.g. "increase of cholesterol level leads to increase of triglycerides level", "increase of age leads to increase of coffee consumption", "increase of education leads to increase of responsibility in the job" or the like. Given the mentioned lack of concrete inter-concept relationships in UMLS, we adopted them, for experimental purposes, as if they were part of this ontology.

2.2 Social Climate: Data and Ontologies

In the second application, both the *ontology*[2] and the *dataset* used for association discovery had the same seed material: the *questionnaire* posed to respondents during the *opinion poll* mapping the 'social climate' of the city of Prague in Spring 2004. The questionnaire contained 51 questions related to e.g. economic situation of families, dwelling, or attitude towards important local events, political parties or media. Some questions consisted of aggregated sub-questions each corresponding to a different 'sign', e.g. "How important is X for you?", where X stands for family, politics, religion etc.; other questions corresponded each to a single 'sign'. The questions were divided into 11 thematic groups.

While the *dataset* was straightforwardly derived from the individual 'signs', each becoming a database column[3], the *ontology* first had the form of *glossary* of candidate terms (manually) picked from the text of the questions; duplicities were removed. In conformance with most ontology engineering methodologies [12], the terms were then divided into candidates for *classes*, *relations* and *instances*, respectively. Then a *taxonomy* and a structure of *non-taxonomic relations* was (again, manually) built, while filling additional entities when needed for better connectivity of the model or just declared as important by domain expert. The instances either correspond to enumerated values of properties, e.g. `GOOD_JOB_AVAILABILITY`, or to outstanding individuals such as `PRAGUE` or `CHRISTIAN_DEMOCRATIC_PARTY`.

The current version of the ontology, formalised in OWL[4], consists of approx. 100 classes, 40 relations and 50 individuals[5]. A Protégé[6] window showing parts of the class hierarchy plus the properties of class `Person` is in Fig. 1. Note that the ambition of our ontology is not to become a widely-usable formal model of social reality; it rather serves for 'simulation' of the possible role of such ontology in the context of KDD.

---

[2] We created the ontology as part of the KDD-oriented project as there was no sufficiently large and rich ontology available in this domain. For a brief overview of existing 'social reality' ontologies see [26].

[3] And, subsequently, an attribute for the *4ft-Miner* tool, see the next section.

[4] `http://www.w3.org/2004/OWL`

[5] By naming convention we adopted, individuals are in capitals, classes start with capital letter (underscore replaces inter-word space for both individuals and classes), and properties start with small letter and the beginning of other than first word is indicated by a capital letter.

[6] `http://protege.stanford.edu`

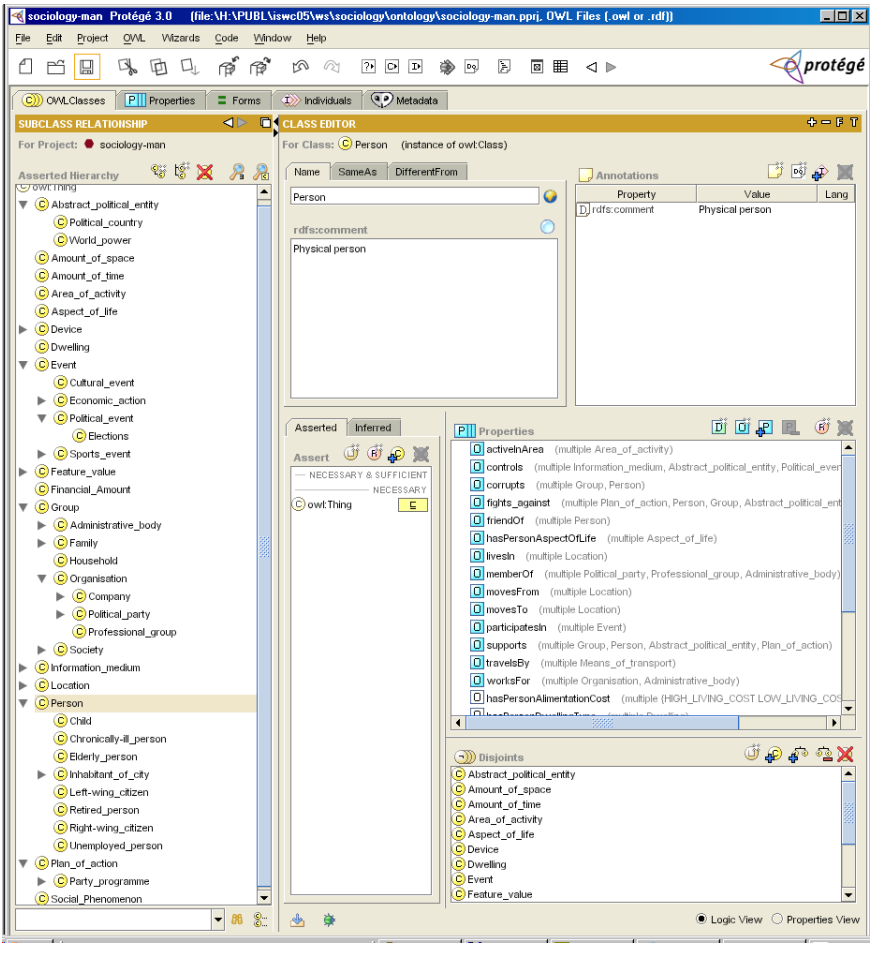


Fig. 1. Incomplete view of the social climate ontology in Protégé

3 Association Mining with *4ft-Miner*

4ft-Miner mines for association rules of the form $\varphi \approx \psi$, where φ and ψ are called *antecedent* and *succedent*, respectively. Antecedent and succedent are conjunctions of *literals*. Literal is a Boolean variable $A(\alpha)$ or its negation $\neg A(\alpha)$, where A is an *attribute* (corresponding to a column in the data table) and α (a set of values called *categories*) is *coefficient* of the literal $A(\alpha)$. The literal $A(\alpha)$ is true for a particular object o in data if the value of A for o is some v such that $v \in \alpha$.

The association rule $\varphi \approx \psi$ means that φ and ψ are associated in the way defined by the symbol $\approx$. The symbol $\approx$, called *4ft-quantifier*, corresponds to a condition over the four-fold contingency table of φ and ψ. The four-fold contingency table of φ and ψ in data matrix $\mathcal{M}$ is a quadruple $\langle a, b, c, d \rangle$ of natural

numbers such that a is the number of data objects from $\mathcal{M}$ satisfying both φ and ψ, b is the number of data objects from $\mathcal{M}$ satisfying φ and not satisfying ψ, c is the number of data objects from $\mathcal{M}$ not satisfying φ and satisfying ψ, and d is the number of from $\mathcal{M}$ from $\mathcal{M}$ satisfying neither φ nor ψ.

There are 16 4ft-quantifiers in *4ft-Miner*. An example of 4ft-quantifier is *above-average dependence*, $\sim^{+}_{p,Base}$, which is defined for $0 < p$ and $Base > 0$ by the condition

$$\frac{a}{a+b} \geq (1+p)\frac{a+c}{a+b+c+d} \wedge a \geq Base \ .$$

The association rule $\varphi \sim^{+}_{p,Base} \psi$ means that among the objects satisfying φ is at least $100p$ per cent more objects satisfying ψ than among all observed objects and that there are at least $Base$ observed objects satisfying both φ and ψ.

As an example of association rule, let us present the expression

$$A(a_1, a_7) \wedge B(b_2, b_5, b_9) \sim^{+}_{p,Base} C(c_4) \wedge \neg D(d_3)$$

Here, $A(a_1, a_7)$, $B(b_2, b_5, b_9)$, $C(c_4)$ and $\neg D(d_3)$ are literals, a_1 and a_7 are categories of A, and $\{a_1, a_7\}$ is the coefficient of $A(a_1, a_7)$[7], and analogously for the remaining literals.

In order to determine the set of relevant questions more easily, we can define *cedents* (i.e. antecedent and/or succedent) φ as a conjunction

$$\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_k$$

where $\varphi_1, \varphi_2, \ldots, \varphi_k$ are *partial cedents*. Each φ_i (itself a conjunction of literals) is chosen from the *set of relevant partial cedents*. The set of partial cedents is given in the following manner:

- the minimum and maximum *length* (i.e. the number of literals in conjunction) of the partial cedent is defined
- a set of *attributes* from which literals will be generated is given
- some attributes can be marked as *basic*, each partial cedent then must contain at least one basic attribute
- a simple definition of the set of all *literals* to be generated is given for each attribute
- *classes of equivalence* can be defined, such that each attribute belongs to at most one class of equivalence; no partial cedent then can contain two or more attributes from the same class of equivalence.

The set of all literals to be generated for a particular attribute is given by:

- the type of coefficient; there are six types of coefficients: subsets, intervals, left cuts, right cuts, cuts[8], one particular category
- the minimum and the maximum length of the literal (in terms of coefficient cardinality)
- positive/negative literal option: only positive, only negative, both.

[7] For convenience, we can write $A(a_1, a_7)$ instead of $A(\{a_1, a_7\})$.

[8] Cuts are intervals containing (at least) one extremal value; cyclical cuts are also possible so as to cover e.g. calendar values.

4 Data Understanding

Within the phase of data understanding, the activity relevant for ontology exploitation is that of *data-to-ontology mapping*, the outcomes of which will be used in later phases.

In the *cardiovascular risk* application we succeeded in mapping 53 of STULONG attributes (from the Entrance dataset) on 19 UMLS semantic types and 25 metathesaurus concepts. Six attributes for which a concept could not be found were only assigned semantic type, for example, 'responsibility in job' was assigned to semantic type Occupational Group. For subsequent processing, we only kept a light-weight fragment of UMLS containing, for each data attribute, the most adequate metathesaurus concept and the least-general semantic type subsuming this concept. We obtained a structure with five taxonomy roots: *Finding*, *Activity*, *Group*, *Food*, and *Disease or Syndrome*.

The side effect of mapping to ontology, as peculiar form of 'data understanding', was occasional identification of *redundant attributes*[9], which (though necessary for data management purposes) were not useful as input to data mining. For example, since the dataset contained the attribute 'age on entrance to STULONG study', the attributes 'birth year' and 'year of entrance to STULONG study' (all mapped to the Age Group semantic type) were of little use.

The mapping between STULONG data and the *qualitative rules* was straightforward, since the data were collected (more-or-less) by the same community of physicians who also formulated the knowledge base, within the same project.

For the same reason, the mapping task was relatively easy in the *social climate* application. Since the core of the ontology had been manually designed based on the text of the questions, mapping amounted to tracking down the links created while building the ontology and maintained during the concept-merging phase. An example of mapping between a question and (fragments of) the ontology is in Fig. 2. Emphasised fragments of the text map to the concepts `Job_availability`, `Metropoly` and `Family` and to the individuals `GOOD_JOB_AVAILABILITY`, `PRAGUE`, `CENTRAL_EUROPE` and `EU`, plus several properties not shown in the diagram. Note that question no. 3 is a 'single-sign' question, i.e. it is directly transformed to one data attribute used for mining. In addition to questions, ontology mapping was also determined for *values* allowed as answers, especially for questions requiring to select concrete objects from a fixed list (city districts, political parties etc.).

5 Task Design

The mining process in narrow sense—*running* an individual mining session—is probably not amenable to ontologies in the case of *4ft-Miner*. The analysis of

[9] Similarly, candidate *missing attributes* could be identified, especially if the concept in the ontology is connected to multiple datatype properties, of which some correspond to mined data attributes and some do not. An algorithm has been suggested for this purpose in [20], though not yet used in the current experiment.

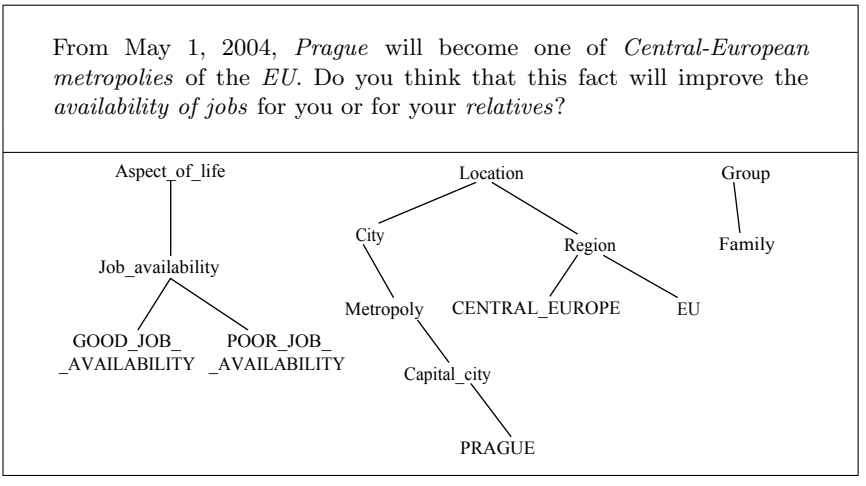


Fig. 2. Question no.3 and fragments of ontology used for its mapping

large data tables relies on optimised database-oriented algorithms, which could hardly accommodate the heterogeneity of ontological information. There is however room for ontologies in the process of *designing* the sessions, due to the sophisticated language for *4ft-Miner* task design (cf. section 3).

In the *cardiovascular risk* application, we used the mapping on ontology concepts from the previous phase so as to identify attributes that should be semantically grouped into partial cedents. We created *partial cedents* covering the attributes mapped on the five upmost classes. Although we carried out this part of the task manually, it could easily be automated.

At a higher level of abstraction, we can also operate on different *task settings*. A very general mining task setting can be decomposed into more specific tasks, which can be run faster, their results will be conceptually more homogeneous, and thus can be interpreted more easily (see below). An example of task decomposition for associations between patient activities and diseases/syndromes is at Fig. 3 (only a few among sub-tasks are included, for illustration). The base task (left branch) might lead to a high number of hypotheses that would be hard to interpret. We can thus e.g. separately refine the antecedent (middle branch) or succedent (right branch) of the base task to obtain more concise and homogeneous results per session.

In the *social climate* application, the ontology was not used in the task design phase. The reason was that the experiments were not guided by the interest of domain experts as in the cardiovascular risk application. So as to allow for the widest possible scope of candidate hypothesis, we thus kept the task definition maximally general: any of 96 attributes (corresponding to 'signs' from the questionnaire) was allowed in antecedent as well as in succedent. As we wanted to start with (structurally) simplest hypotheses, we set the length of antecedent as well as of succedent to 1, and the cardinality of coefficient also to 1 (i.e., choice of

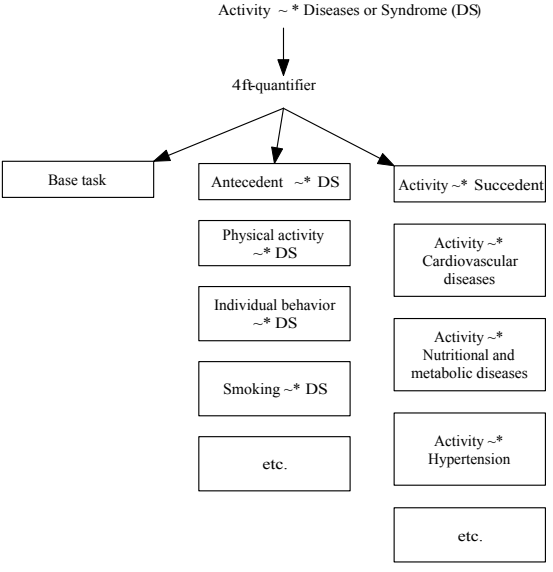


Fig. 3. Decomposition of 4ft tasks with respect to ontology

single category). As quantifier we used the *above-average dependence* explained in section 3. The run-times were typically lower than a second.

6 Result Interpretation

Given the data-to-ontology mapping, concrete associations discovered with the help of *4ft-Miner* can be matched to corresponding semantic relations or their more complex chains from the ontology, see Fig. 4. The semantic relation represents a potential context (e.g. explanation) for the discovered association.

In the *cardiovascular risk* application, each mining task already corresponded to a meaningful 'framing' question, such as "Searching for relationships between

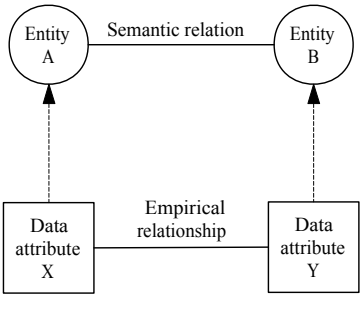


Fig. 4. Semantic relation as context to empirical relationship between attributes

Table 1. Four-fold table for a 'confirmation' association

	Succedent	NOT Succedent	
Antecedent	10	22	32
NOT Antecedent	66	291	357
	76	313	389

Activity of the patient and Diseases/Syndromes". Concrete associations discovered with the help of *4ft-Miner* could then be compared with the instance layer of the ontology; in our case, with the qualitative rules. The relationship of an association with prior knowledge is typically one of the following:

- Confirmation of prior knowledge, without additional information
- New knowledge compatible with prior knowledge
- Exception to or conflict with prior knowledge.

Let us show two examples, with their four-fold tables:

- The discovered association "Patients who are not physically active within the job nor after the job (Antecedent) will more often have higher blood pressure (Succedent)" was derivable from the field knowledge (qualitative rule) "Patients who are physically active after the job will more often have lower blood pressure" (Table 1).
- The discovered association "94% of patients smoking 5 or more cigarettes a day for more than 21 years (Antecedent) have neither myocardial infarction nor ictus nor diabetes (Succedent)" was in conflict with prior knowledge "Increase of smoking leads to increase of cardio-vascular diseases" (Table 2).

The examples are merely illustrative. In order to draw medically valid conclusions from them, we would at least need to examine the statistical validity of the hypotheses in question. As the STULONG dataset is relatively small, few such hypotheses actually pass conventional statistical tests.

In the *social climate* application, associations are potentially even harder to interpret than in the medical domain, as the attributes correspond to somewhat ad hoc statements rather than to established quantities, measurements and criteria. Furthermore, we did not have concrete field knowledge at our disposal. We thus used the ontology itself—not to directly compare it with the hypotheses but to retrieve entity (concept-relation) chains that could serve as templates

Table 2. Four-fold table for a 'conflicting' association

	Succedent	NOT Succedent	
Antecedent	216	14	230
NOT Antecedent	145	14	159
	361	28	389

for candidate *explanations* of the hypotheses. Again, we did not have an appropriate software support for extracting entity chains (i.e. explanation templates) from the ontology, and only examined it via manual browsing. As a side-effect of chain extraction, we also identified *missing* (though obvious) links among the classes, which could be added to the ontology, and also some modelling *errors*, especially, domain/range constraints at an inappropriate level of generality.

We divided the strong hypotheses resulting from *4ft-Miner* runs into four groups, with respect to their amenability to ontology-based explanation:

1. *Strict dependencies*, e.g. the association between answers to the questions "Do you use a public means of transport?" and "Which public means of transport do you use?". Such results are of no interest in KDD and could of course be eliminated with more careful task design.
2. Relationships amounting to *obvious causalities*, for example, the association between "Are you satisfied with the location where you live?" and "Do you intend to move?" Such relationships (in particular, their strengh) might be of some interest for KDD in general; however, there is no room for ontology-based explanation, since both the antecedent and succedent are mapped on the same or directly connected ontology concepts (`Location`, `livesIn`, `movesFrom` etc.).
3. Relationships between signs that have the character of respondent's agreement with relatively *vague propositions*, for example "Our society changes too fast for a man to follow." and "Nobody knows what direction the society is taking." We could think of some complex ontology relationships, however, by Occam's razor, it is natural just to assume that the explanation link between the antecedent and succedent goes through the categorisation of the respondent as conservative/progressist or the like.
4. Relationships between signs corresponding to concrete and relatively *semantically distant* questions (in fact, those appearing in different thematic groups in the questionnaire). This might be e.g. the question "Do you expect that the standard of living of most people in the country will grow?", with answer 'certainly not', and the question "Which among the parties represented in the city council has a programme that is most beneficial for Prague?" with 'KSČM' (the Czech Communist Party) as answer. Such *cross-group* hypotheses are often amenable to ontology-based explanation.

The last hypothesis mentioned, formally written as `Z05(4)` $\sim^{+}_{0.22,64}$ `Z18(3)`, can be visualised in *4ft-Miner* by means of *four-fold contingency table*, as shown at Fig. 5, and also graphically (see [26]). The contingency table (followed with a long list of computed characteristics) shows that:

- 64 people disagree that the standard of living would grow AND prefer KSČM
- 224 people disagree that the standard of living would grow AND DO NOT prefer KSČM
- 171 people DO NOT disagree[10] that the standard of living would grow AND prefer KSČM

[10] More precisely, their answer to the question above was not 'certainly not'; it was one of 'certainly yes', 'probably yes', 'probably no'.

– 2213 people DO NOT disagree that the standard of living would grow AND DO NOT prefer KSČM.

We can see that among the people who disagree that the standard of living would grow, there is a 'substantially' higher number of people who also prefer KSČM than in the whole data sample, and vice versa[11]. The whole effort of formulating hypotheses about the reason for this association is however on the shoulders of the human expert.

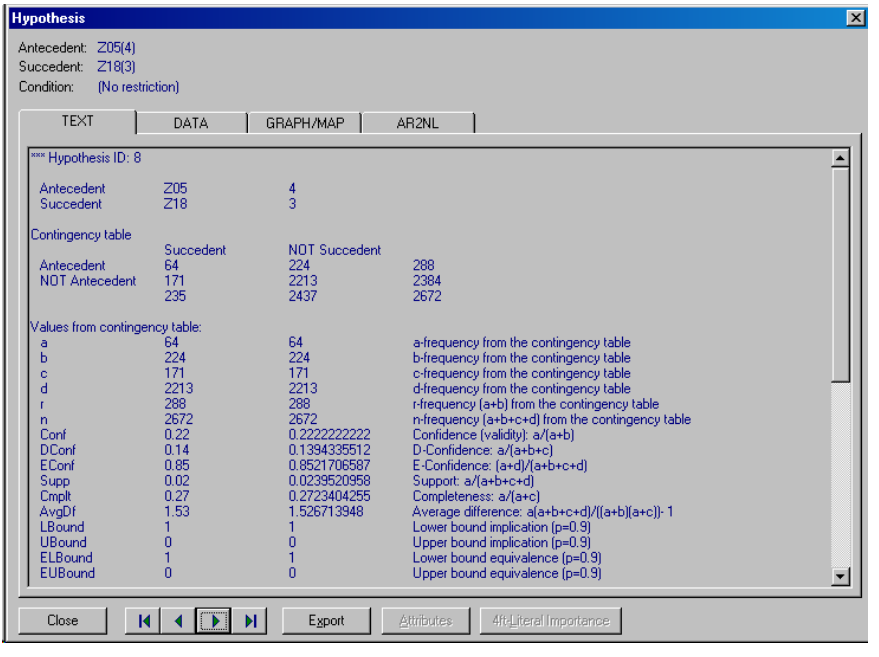


Fig. 5. Textual view of a *4ft-Miner* hypothesis

In order to identify potential *explanation templates*, we took advantage of the *mapping* created prior to the knowledge discovery phase, see section 4. The negative answer to the question about standard of living was mapped to the individual `BAD_LIVING_STANDARD` (instance of `Social_phenomenon`), and the respective answer to the question about political parties was mapped to the class `Political_party`, to its instance `KSCM`, to the class `Party_programme` and to the class `City_council`. Table 3 lists some among the possible templates, first ordered by the decreasing number of involved entities on which the hypothesis is *mapped* and then by the decreasing number of *all* involved entities. The templates do not contain intermediate classes from the hierarchy (which are not even counted for the ordering). Relations are only considered as linked to the class for which they are directly defined as domain/range, i.e. not to the class

[11] This is the principle of the *above-average* quantifier, which is symmetrical.

Table 3. Explanation templates for 'standard of living' vs. 'KSČM' association

Template	Mapped	All
KSCM ∈ **Political_party** hasPartyProgramme **Party_programme** ⊑ Plan_of_action hasObjective Social_phenomenon ∋ **BAD_LIVING_STANDARD**	4	6
KSCM ∈ **Political_party** isRepresentedIn Administrative_body ⊒ **City_council** carriesOutAction Economic_action hasImpactOn Social_phenomenon ∋ **BAD_LIVING_STANDARD**	4	7
KSCM ∈ **Political_party** hasPartyProgramme **Party_programme** ⊑ Plan_of_action envisagesAction Action ⊒ Economic_action hasImpactOn Social_phenomenon ∋ **BAD_LIVING_STANDARD**	4	8
KSCM ∈ Group informsAbout Social_phenomenon ∋ **BAD_LIVING_STANDARD**	2	3
KSCM ∈ Group carriesOut Action ⊒ Economic_action hasImpactOn Social_phenomenon ∋ **BAD_LIVING_STANDARD**	2	6
KSCM ∈ Group participatesIn Event ⊒ Economic_action hasImpactOn Social_phenomenon ∋ **BAD_LIVING_STANDARD**	2	6
KSCM ∈ Group supports Action ⊒ Economic_action hasImpactOn Social_phenomenon ∋ **BAD_LIVING_STANDARD**	2	6
KSCM ∈ Group fightsAgainst Group carriesOutAction Action ⊒ Economic_action hasImpactOn Social_phenomenon ∋ **BAD_LIVING_STANDARD**	2	7

that inherits them. The symbols ⊑, ⊒ stand for subclass/superclass relationship and ∈, ∋ for instance-to-class membership[12].

We can see that the 'most preferable' template suggests that the KSČM party may have some programme that may have as objective to reach the phenomenon of BAD_LIVING_STANDARD. The second looks a bit more adequate: the KSČM party is represented in the city council that can carry out an economic action that may have some impact on the phenomenon of BAD_LIVING_STANDARD. The third is almost identical to the first one. The fourth (and simplest) might actually be most plausible: the KSČM party informs about the phenomenon of BAD_LIVING_STANDARD. Let us finally mention the fifth template, which builds on an incorrect 'inference' (caused by imprecise modelling): the party is assumed to carry out an economic action, which it (directly) can't. The relation was defined with Group and Action as subsets of its domain and range, respectively. However, the combination of Political_party (subclass of Group)

[12] Note that the description-logic-like notation is only used for brevity; a more user-oriented (e.g. graphical) representation would probably be needed to provide support for a domain expert not familiar with knowledge representation conventions.

and `Economic_action` (subclass of `Action`) is illegal and should have been ruled out by an axiom such as `Political_party` ⊒ `(ALL carriesOutAction (NOT Economic_action))`.

7 Result Deployment over Semantic Web

The role of ontology in the deployment phase is most crucial if the mining results are to be supplied to a wider (possibly unrestricted) range of consumer applications. A promising approach would be to incorporate the mining results into *semantic web* documents. The most straightforward way to do so is to take advantage of *analytic reports* (ARs): textual documents presenting the results of KDD process in a condensed form. ARs are produced by humans, although the use of natural language generation was also studied [24]. Prior to entering the reports, the sets of discovered hypotheses, understood as formulae in the so-called observational calculus, can be transformed using formal *deduction rules* [21,22] into 'canonical form' (which is, among other, free of redundancies). In the *cardiovascular risk* application, a collection of ARs has been created by junior researchers based upon results of selected *4ft-Miner* tasks on STULONG data. Similarly, in the *social climate* application, an almost exhaustive collection of (about 60) ARs have been created for different task settings (combinations of attribute groups), by undergraduate students as part of their assignment.

In order to embed the formal representation of *4ft-Miner* results themselves into the text of the reports [16], we initially used an original XML-based language conforming to early RuleML specifications [1]. A more up-to-date option would be to combine such rules with an ontology (mapped on data attributes, cf. section 4), as proposed e.g. by the Semantic Web Rule Language [15]. However, we also consider another option, which would go in the spirit of ontology learning [8,17]: to use association rule mining to learn (skeletons of) *OWL ontologies* from data. The knowledge contained in the analytic reports would then be represented as ontology axioms rather than as rules, which would enable us to exploit description logic reasoners to formally compare the sets of results.

The decision whether to replace rules with OWL axioms in modelling logical implications would probably be based on two aspects: (1) the number of variables in the rule: if there is only one variable then the expression can usually be expressed in OWL as concept subsumption; (2) the nature of the implication: if it has 'conceptual' nature then it should be modelled in OWL if possible, while if it is 'ad hoc' (say, purely empirical) then rules might be a better choice. The first aspect makes OWL a preferable choice for our case, providing the association mining is carried out on a single data table. Then the resulting taxonomy is subordinated to a single ontology node, such as Patient (in the cardiovascular risk application) or Citizen_of_Prague (in the social climate application). The initial domain ontology then can be used to 'unfold' some concepts in the taxonomy into restrictions over object properties. For example, from the associations discovered in the sociological domain, we can construct taxonomy path such as Citizen_of_Prague ⊒ KSČM_supporter ⊒ Inhabitant_of_District_14

$\sqsupseteq$ Citizen_wishing_to_move_to_District_15. We can then unfold the concept of ODS_supporter to a 'to-value' restriction Inhabitant_of_District_14 $\sqsubseteq$ (supports $\in$ KSČM). In order to preserve the information content of the original hierarchy, unfolding should not be carried out for both the parent and child concept, i.e., at most every other concept along the path can be left out.

As the associations are equipped with confidence factors, we should consider some formalism for modelling impreciseness, such as fuzzy versions of OWL.

8 Envisaged Tool Support

For the modelling (i.e. mining) phase of experiments, the version of *4FT-Miner* (cf. section 3) implemented in the *LISp-Miner* tool was used. *LISp-Miner*[13] is a robust and scalable system, which also includes five other mining procedures in addition to *4ft-Miner*. However, since 2002 there has been a new system under development named *Ferda*[14]. The authors of *Ferda* aimed to create an open user-friendly system, based on the long-term experience of *LISp-Miner*, but relying on principles of visual programming. For example, Fig. 6 shows the *Ferda* visual environment with the setting of a task examining the validity of qualitative rule "increase of cholesterol leads to increase of triglycerides level" mentioned in Section 2.1. Another important feature of *Ferda* is its extensibility. The user can easily add a new module to the system, which can communicate with other modules via predefined interfaces. [14] describes *Ferda* in more detail. Because of the extensibility and available implementation of GUHA procedures in *Ferda*, it is highly preferable to implement new tools connecting ontologies and association mining in this system. Currently, it is possible in *Ferda* to construct and validate qualitative rules against hypotheses generated by data mining runs. Furthermore, proposals for modules exploiting ontologies for task setup (automatic identification of redundant attributes, automatic categorisation of attributes, and even for automated setup of the whole task) have been formulated in [20].

9 Related Work

Although domain ontologies are a popular instrument in many diverse applications incl. e.g. text minig, they only scarcely appeared in 'tabular' KDD until very recently. A notable exception was the work by Philips & Buchanan [19], where 'common-sense' ontologies of time and processes were exploited to derive constraints on attributes, which were in turn used to construct new attributes; this is somewhat analogous (though not identical) to the detection of missing attributes mentioned in section 4. Although not explicitly talking about ontologies, the work by Clark & Matwin [10] is also relevant; they used qualitative models as bias for inductive learning (i.e. in the sense of our section 5). Finally, Thomas et al. [27]

[13] `http://lispminer.vse.cz`

[14] *Ferda* can be downloaded at `http://ferda.sourceforge.net`

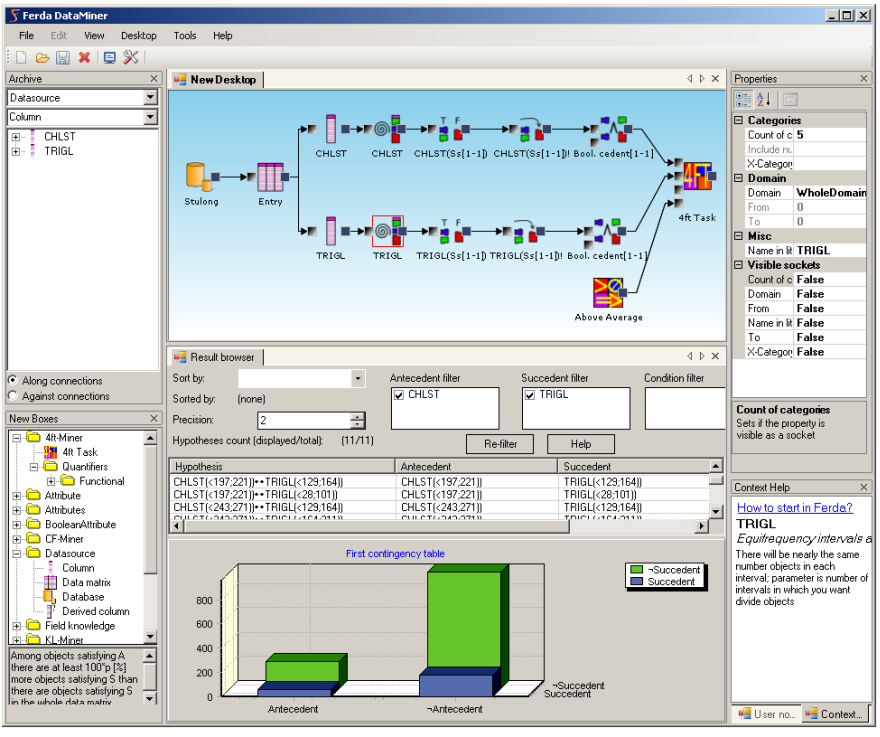


Fig. 6. The *Ferda* environment

and van Dompseler & van Someren [28] used problem-solving method descriptions (a kind of 'method ontologies') for the same purpose. There have also been several efforts to employ taxonomies over domains of individual attributes [3,4,18,25] to guide inductive learning. A recent contribution that goes in similar direction with our work on hypothesis interpretation but is more restricted in scope is that of Domingues&Rezende [11], which uses ontologies (namely, taxonomies) to post-process the results of association mining via generalisation and pruning. Finally, a specific stream of research is represented by bioinformatics applications that exploit (usually, shallow) ontologies in mining gene data, see e.g. [6,7].

10 Conclusions and Future Work

We presented a pilot study on using ontologies to enhance the knowledge discovery process; the study was carried along most phases of the process (data understanding, task design, result interpretation, deployment) and targeted into two applications: cardiovascular risk and social climate. The KDD task examined is association mining. For some of the ontology-related tasks, existing software (such as the *Ferda* tool) can be used or adapted.

The study discovered a large and heterogeneous collection of entry points for ontologies in the KDD process. Some of them can be exploited straightforwardly while other have numerous pre-requisites; some are almost guaranteed to improve the process while for others the effects are unsure to unweigh the costs. In the future, we need to invest more theoretical as well as engineering effort in particular to data-ontology mapping and subsequent *matching among discovered hypotheses and ontology chains*. This task has no software support at the moment; the added value of ontologies is potentially very high here, while the assumptions taken in this study are a bit strong and may not hold in other settings. In a longer run, it would also be desirable to extend the scope of the project towards discovered hypotheses with *more complex structure*, e.g. with longer antecedents/succedents, with additional condition, or even to hypotheses discovered by means of a different procedure than *4ft-Miner*.

Acknowledgements

The research is partially supported by the grant no.201/05/0325 of the Czech Science Foundation, "New methods and tools for knowledge discovery in databases". We acknowledge the contribution of our research colleagues and domain experts to partial results, namely, of Hana Češpivová, Miroslav Flek, Martin Kejkula and Marie Tomečková, and valuable comments from the anonymous reviewers.

References

1. The Rule Markup Initiative, `http://www.ruleml.org/`.
2. Unified Medical Language System, `http://www.nlm.nih.gov/research/umls/`.
3. Almuallim, H., Akiba, Y. A., Kaneda, S.: On Handling Tree-Structured Attributes in Decision Tree Learning. In: Proc. ICML 2005, Morgan Kaufmann, 12–20.
4. Aronis, J.M., Provost, F.J., Buchanan, B.G.: Exploiting Background Knowledge in Automated Discovery. In: Proc. SIGKDD-96.
5. Berendt, B., Hotho, A., Stumme, G.: 2nd Workshop on Semantic Web Mining, held at ECML/PKDD-2002, Helsinki 2002, `http://km.aifb.uni-karlsruhe.de/semwebmine2002`.
6. Cannataro, M., Guzzi, P. H., Mazza, T., Tradigo, G., Veltri, P.: Using Ontologies in PROTEUS for Modeling Proteomics Data Mining Applications. In: From Grid to Healthgrid: Proceedings of Healthgrid 2005, IOS Press, 17-26.
7. Brisson, L., Collard, M., Le Brigant, K., Barbry, P.: KTA: A Framework for Integrating Expert Knowledge and Experiment Memory in Transcriptome Analysis. In: International Workshop on Knowledge Discovery and Ontologies, held with ECML/PKDD 2004, Pisa, p.85-90.
8. Buitelaar, P., Cimiano, P., Magnini, B. (eds.): Ontology Learning and Population, IOS Press, 2005.
9. Češpivová, H., Rauch, J., Svátek V., Kejkula M., Tomečková M.: Roles of Medical Ontology in Association Mining CRISP-DM Cycle. In: ECML/PKDD04 Workshop on Knowledge Discovery and Ontologies (KDO'04), Pisa 2004.
10. Clark, P. Matwin, S.: Using Qualitative Models to Guide Inductive Learning. In: Proceedings of the 1993 International Conference on Machine Learning, 49-56.

11. Domingues, M. A., Rezende S. A.: Using Taxonomies to Facilitate the Analysis of the Association Rules. In: The 2^{nd} International Workshop on Knowledge Discovery and Ontologies, held with ECML/PKDD 2005, Porto, p.59-66.
12. Gómez-Perez, A., Fernández-Lopez, M., Corcho, O.: Ontological Engineering. Springer 2004.
13. Hájek, P., Havránek, T.: *Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory*. Springer-Verlag: Berlin - Heidelberg - New York, 1978.
14. Kováč, M., Kuchař, T., Kuzmin A.: Ferda, New Visual Environment for Data Mining (in Czech). In: Znalosti 2006, Czecho-Slovak Knowledge Technology Conference, Hradec Králové 2006, 118–129.
15. Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Submission, 21 May 2004. Online `http://www.w3.org/Submission/SWRL`.
16. Lín, V., Rauch, J., Svátek, V.: Content-based Retrieval of Analytic Reports. In: Schroeder, M., Wagner, G. (eds.). Rule Markup Languages for Business Rules on the Semantic Web, Sardinia 2002, 219–224.
17. Maedche, A.: Ontology Learning for the Semantic Web. Kluwer, 2002.
18. Núñez, M.: The Use of Background Knowledge in Decision Tree Induction. *Machine Learning*, 6, 231–250 (1991).
19. Phillips, J., Buchanan, B.G.: Ontology-guided knowledge discovery in databases. In: International Conf. Knowledge Capture, Victoria, Canada, 2001.
20. Ralbovský M.: Usage of Domain Knowledge for Applications of GUHA Procedures (in Czech), Master thesis, Faculty of Mathematics and Physics, Charles University, Prague 2006.
21. Rauch, J.: Logical Calculi for Knowledge Discovery in Databases. In: Principles of Data Mining and Knowledge Discovery (PKDD-97), Springer-Verlag, 1997.
22. Rauch, J.: Logic of Association Rules. *Applied Intelligence*, 22, 9-28, 2005.
23. Rauch, J., Šimůnek, M.: An Alternative Approach to Mining Association Rules. In: Lin, T. Y., Ohsuga, S., Liau, C. J., Tsumoto, S. (eds.), Data Mining: Foundations, Methods, and Applications, Springer-Verlag, 2005, pp. 211–232
24. Strossa, P., Černý, Z., Rauch, J.:. Reporting Data Mining Results in a Natural Language. In: Lin, T. Y., Ohsuga, S., Liau, C. J., Hu, X. (ed.): Foundations of Data Mining and Knowledge Discovery. Berlin : Springer, 2005, pp. 347-362
25. Svátek, V.: Exploiting Value Hierarchies in Rule Learning. In: van Someren, M. - Widmer, G. (Eds.): ECML'97, 9th European Conference on Machine Learning. Poster Papers. Prague 1997, 108–117.
26. Svátek, V., Rauch, J., Flek, M.: Ontology-Based Explanation of Discovered Associations in the Domain of Social Reality. In: The 2^{nd} ECML/PKDD Workshop on Knowledge Discovery and Ontologies, 2005, Porto, 75-86.
27. Thomas J., Laublet, P., Ganascia, J. G.: A Machine Learning Tool Designed for a Model-Based Knowledge Acquisition Approach. In: EKAW-93, European Knowledge Acquisition Workshop, Lecture Notes in Artificial Intelligence No.723, N.Aussenac et al. (eds.), Springer-Verlag, 1993, 123–138.
28. van Dompseler, H. J. H., van Someren, M. W.: Using Models of Problem Solving as Bias in Automated Knowledge Acquisition. In: ECAI'94 - European Conference on Artificial Intelligence, Amsterdam 1994, 503–507.

A Website Mining Model Centered on User Queries

Ricardo Baeza-Yates[1,2,3] and Barbara Poblete[1,2]

[1] Web Research Group, Technology Department,
University Pompeu Fabra, Barcelona, Spain
[2] Center for Web Research, CS Department
University of Chile, Santiago, Chile
[3] Yahoo! Research, Barcelona, Spain
{ricardo.baeza, barbara.poblete}@upf.edu

Abstract. We present a model for mining user queries found within the access logs of a website and for relating this information to the website's overall usage, structure and content. The aim of this model is to discover, in a simple way, valuable information to improve the quality of the website, allowing the website to become more intuitive and adequate for the needs of its users. This model presents a methodology of analysis and classification of the different types of queries registered in the usage logs of a website, such as queries submitted by users to the site's internal search engine and queries on global search engines that lead to documents in the website. These queries provide useful information about topics that interest users visiting the website and the navigation patterns associated to these queries indicate whether or not the documents in the site satisfied the user's needs at that moment.

1 Introduction

The Web has been characterized by its rapid growth, massive usage and its ability to facilitate business transactions. This has created an increasing interest for improving and optimizing websites to fit better the needs of their visitors. It is more important than ever for a website to be found easily in the Web and for visitors to reach effortlessly the contents they are looking for. Failing to meet these goals can result in the loss of many potential clients.

Web servers register important data about the usage of a website. This information generally includes visitors navigational behavior, the queries made to the website's internal search engine (if one is available) and also the queries on external search engines that resulted in requests of documents from the website, queries that account for a large portion of the visits of most sites on the Web. All of this information is provided by visitors implicitly and can hold the key to significantly optimize and enhance a website, thus improving the "quality" of that site, understood as *"the conformance of the website's structure to the intuition of each group of visitors accessing the site"* [1].

Most of the queries related to a website represent actual information needs of the users that visit the site. However, user queries in Web mining have been

M. Ackermann et al. (Eds.): EWMF/KDO 2005, LNAI 4289, pp. 1–17, 2006.

studied mainly with the purpose of enhancing website search, and not with the intention of discovering new data to increase the quality of the website's contents and structure. For this reason in this paper we present a novel model that mines queries found in the usage logs of a website, classifying them into different categories based in navigational information. These categories differ according to their importance for discovering new and interesting information about ways to improve the site. Our model also generates a visualization of the site's content distribution in relation to the link organization between documents, as well as the URLs selected due to queries. This model was mostly designed for websites that register traffic from internal and/or external search engines, even if this is not the main mechanism of navigation in the site. The output of the model consists of several reports from which improvements can be made to the website.

The main contributions of our model for improving a website are: *to mine user queries within a website's usage logs*, *obtain new interesting contents to broaden the current coverage of certain topics in the site*, *suggest changes or additions to words in the hyperlink descriptions*, and at a smaller scale *suggest to add new links between related documents and revise links between unrelated documents in a site*.

We have implemented this model and applied it to different types of websites, ranging from small to large, and in all cases the model helps to point out ways to improve the site, even if this site does not have an internal search engine. We have found our model specially useful on large sites, in which the contents have become hard to manage for the site's administrator.

This paper is organized as follows. Section 2 presents related work and section 3 our model. Section 4 gives an overview of our evaluation and results. The last section presents our conclusions and future work.

2 Related Work

Web mining [2] is the process of discovering patterns and relations in Web data. Web mining generally has been divided into three main areas: *content mining*, *structure mining* and *usage mining*. Each one of these areas are associated mostly, but not exclusively, to these three predominant types of data found in a website:

Content: The "real" data that the website was designed to give to its users. In general this data consists mainly of text and images.

Structure: This data describes the organization of the content within the website. This includes the organization inside a Web page, internal and external links and the site hierarchy.

Usage: This data describes the use of the website, reflected in the Web server's access logs, as well as in logs for specific applications.

Web usage mining has generated a great amount of commercial interest [3,4]. The analysis of Web server logs has proven to be valuable in discovering many

issues, such as: if a document has never been visited it may have no reason to exist, or on the contrary, if a very popular document cannot be found from the top levels of a website, this might suggest a need for reorganization of its link structure.

There is an extensive list of previous work using Web mining for improving websites, most of which focuses on supporting adaptive websites [5] and automatic personalization based on Web Mining [6]. Amongst other things, using analysis of frequent navigational patterns and association rules, based on the pages visited by users, to find interesting rules and patterns in a website [1,7,8,9,10]. Other research targets mainly modeling of user sessions, profiles and cluster analysis [11,12,13,14,15].

Queries submitted to search engines are a valuable tool for improving websites and search engines. Most of the work in this area has been directed at using queries to enhance website search [16] and to make more effective global Web search engines [17,18,19,20]. In particular, in [21] chains (or sequences) of queries with similar information needs are studied to learn ranked retrieval functions for improving Web search. Queries can also be studied to improve the quality of a website. Previous work on this subject include [22] which proposed a method for analyzing similar queries on Web search engines, the idea is to find new queries that are similar to ones that directed traffic to a website and later use this information to improve the website. Another kind of analysis based on queries, is presented in [23] and consists of studying queries submitted to a site's internal search engine, and indicates that valuable information can be discovered by analyzing the behavior of users in the website after submitting a query. This is the starting point of our work.

3 Model Description

In this section we will present the description of our model for mining website usage, content and structure, centered on queries. This model performs different mining tasks, using as input the website's access logs, its structure and the content of its pages. These tasks also includes data cleaning, session identification, merging logs from several applications and removal of robots amongst other things which we will not discuss in depth at this moment, for more details please refer to [24,25,26]. The following concepts are important to define before presenting our model:

Session: A session is a sequence of document accesses registered for one user in the website's usage logs within a maximum time interval between each request. This interval is set by default to 30 minutes, but can be changed to any other value considered appropriate for a website [24]. Each user is identified uniquely by the `IP` and `User-Agent`.

Queries: A query consists of a set of one or more keywords that are submitted to a search engine and represents an information need of the user generating that query.

Information Scent: IS [27] indicates how well a word, or a set of words, describe a certain concept in relation to other words with the same semantics. For example, polysemic words (words with more than one meaning) have less IS due to their ambiguity.

In our model the structure of the website is obtained from the links between documents and the content is the text extracted from each document. The aim of this model is to generate information that will allow to improve the structure and contents of a website, and also to evaluate the interconnections amongst documents with similar content.

For each query that is submitted to a search engine, a page with results is generated. This page has links to documents that the search engine considers appropriate for the query. By reviewing the brief abstract of each document displayed (which allows the user to decide roughly if a document is a good match for his or her query) the user can choose to visit zero or more documents from the results page. Our model analyzes two different types of queries, that can be found in a website's access registries. These queries are:

External queries: These are queries submitted on Web search engines, from which users selected and visited documents in a particular website. They can be discovered from the log's `referer` field.

Internal queries: These are queries submitted to a website's internal search box. Additionally, external queries that are specified by users for a particular site, will be considered as internal queries for that site. For example, Google.com queries that include `site:example.com` are internal queries for the website `example.com`. In this case we can have queries without clicked results.

Figure 1 (left) shows the description of the model, which gathers information about internal and external queries, navigational patterns and links in the website to discover IS that can be used to improve the site's contents. Also the link and content data from the website is analyzed using clustering of similar documents and connected components. These procedures will be explained in more detail in the following subsections.

3.1 Navigational Model

By analyzing the navigational behaviors of users within a website, during a period of time, the model can classify documents into different types, such as: *documents reached without a search*, *documents reached from internal queries* and *documents reached from external queries*. We define these types of documents as follows:

Documents reached Without a Search (DWS): These are documents that, throughout the course of a session, were reached by browsing and without the interference of a search (in a search engine internal or external to the website). In other words, documents reached from the results page of a search

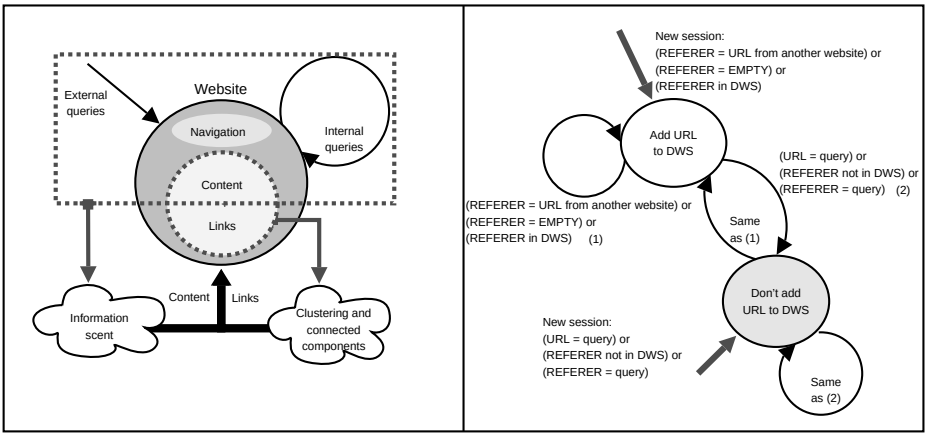


Fig. 1. Model description *(left)* and heuristic for DWS *(right)*

engine and documents attained from those results, are *not* considered in this category. Any document reached from documents visited previously to the use of a search engine will be considered in this category.

Documents reached from Internal Queries (DQ_i): These are documents that, throughout the course of a session, were reached by the user as a direct result of an *internal query*.

Documents reached from External Queries (DQ_e): These are documents that, throughout the course of a session, were reached by the user as a direct result of an *external query*.

For future references we will drop the subscript for DQ_i and DQ_e and will refer to these documents as DQ.

It is important to observe that DWS and DQ are *not disjoint sets of documents*, because in one session a document can be reached using a search engine (therefore belonging to DQ) and in a different session it can also be reached without using a search engine. The important issue then, is to register *how many times* each of these different events occur for each document. We will consider the frequency of each event directly proportional to that event's significance for improving a website. The classification of documents into these three categories will be essential in our model for discovering useful information from queries in a website.

Heuristic to Classify Documents. Documents belonging to DQ sets can be discovered directly by analyzing the referer URL in an HTTP request to see if it is equal to the results page of a search engine (internal or external). In these cases only the *first occurrence* of each requested document in a session is classified. On the other hand, documents in DWS are more difficult to classify, due to the fact that backward and forward navigation in the browser's cached history of previously visited documents is not registered in web servers usage

logs. To deal with this issue we created the heuristic shown in Figure 1, which is supported by our empirical results. Figure 1 (right) shows a state diagram that starts a new classification at the beginning of each session and then processes sequentially each request from the session made to the website's server. At the beginning of the classification the set DWS is initialized to the value of the website's start page (or pages) and any document requested from a document in the DWS set, from another website or from an empty referer (the case of bookmarked documents) are added to the DWS set.

3.2 Query Classification

We define different types of queries according to the outcome observed in the user's navigational behavior within the website. In other words, we classify queries in relation to: if the user chooses to visit the generated results and if the query had results in the website. Our classification can be divided into two main groups: *successful queries* and *unsuccessful queries*. Successful queries can be found both in internal and external queries, but unsuccessful queries can only be found for internal queries since all external queries in the website's usage logs were successful for that site.

Successful Queries. If a query submitted during a session had visited results in that same session, we will consider it as a successful query. There are two types of successful queries, which we will call A and B. We define formally classes A and B queries as follows (see Figure 2):

Class A queries: Queries for which the session visited one or more results in *AD*, where AD contains documents found in the DWS set. In other words, the documents in AD have also been reached, in at least one other session, browsing without using a search engine.

Class B queries: Queries for which the session visited one or more results in *BD*, where BD contains documents that are only classified as DQ and not in DWS. In other words documents in BD have *only* been reached using a search in all of the analyzed session.

The purpose of defining these two classes of queries, is that A and B queries *contain keywords that can help describe the documents that were reached as a result of these queries*. In the case of A queries, these keywords can be used in the text that describes links to documents in AD, contributing additional IS for the existing link descriptions to these documents. The case of B queries is even more interesting, because the words used for B queries describe documents in BD better than the current words used in link descriptions to these documents, contributing with new IS for BD documents. Also, the most frequent documents in BD should be considered by the site's administrator as good suggestions of documents that should be reachable from the top levels in the website (this is also true in minor extent for AD documents). That is, we suggest hotlinks based on queries and not on navigation, as is usual. It is important to consider that the same query can co-occur in class A and class B (what cannot co-occur is

the same document in AD and BD!), so the relevance associated to each type of query is proportional to its frequency in each one of the classes in relation to the frequency of the document in AD or BD.

Unsuccessful Queries. If a query submitted to the internal search engine did not have visited results in the session that generated it, we will consider it as an unsuccessful query. There are two main causes for this behavior:

1. The search engine displayed zero documents in the results page, because there were no appropriate documents for the query in the website.
2. The search engine displayed one or more results, but none of them seemed appropriate from the user's point of view. This can happen when there is poor content or with queries that have polysemic words.

There are four types of unsuccessful queries, which we will call C, C', D and E. We define formally these classes of queries as follows (see Figure 2):

Class C queries: Queries for which the internal search engine displayed results, but the user choose not no visit them, probably because there were no appropriate documents for the user's needs at that moment. This can happen for queries that have ambiguous meanings and for which the site has documents that reflect the words used in the query, but not the concept that the user was looking for. It can also happen when the contents of the site do not have the specificity that the user is looking for. Class C queries represent concepts that should be developed in depth in the contents of the website with the meaning that users intended, focused on the keywords of the query.

Class C' queries: Queries for which the internal search engine did not display results. This type of query requires a manual classification by the webmaster of the site. If this manual classification establishes that the concept represented by the query *exists* in the website, but described with different words, then this is a class C' query. These queries represent words that should be used in the text that describes links and documents that share the same meaning as these queries.

Class D queries: As in class C' queries, the internal search engine did not display results and manual classification is required. However, if in this case, the manual classification establishes that the concept represented by the query does *not exist* in the website, but we believe that it should appear in the website, then the query is classified as class D. Class D queries represent concepts that should be included in documents in the website, because they represent new topics that are of interest to users of the website.

Class E queries: Queries that are not interesting for the website, as there are no results, but it's not a class C' or class D query, and should be omitted in the classification[1].

[1] This includes typographical errors in queries, which could be used for a hub page with the right spelling and the most appropriate link to each word.

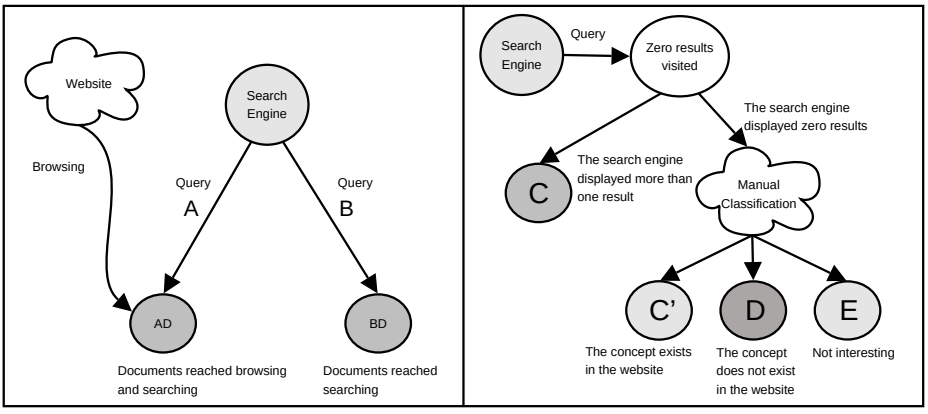


Fig. 2. Successful queries *(right)* and unsuccessful queries *(left)*

Table 1. Classes of queries and their contribution to the improvement of a website

Class	Concept exists	Results displayed	Visited documents	Significance	Contribution	Affected component
A	yes	yes	DQ ∩ DWS	low	additional IS	anchor text
B	yes	yes	DQ \ DWS	high	new IS, add hotlinks	anchor text, links
C	yes	yes	∅	medium	new content	documents
C'	yes	no	—	medium	new IS	anchor text, documents
D	no, but it should	no	—	high	new content	anchor text, documents
E	no	no	—	none	—	—

Each query class is useful in a different way for improving the website's content and structure. The importance of each query will be considered proportional to that query's frequency in the usage logs, and each type of query is only counted once for every session. Table 1 shows a review of the different classes of queries.

Manual classification is assisted by a special interface in our prototype implementation. The classification is with memory (that is, an already classified query does not need to be classified in a subsequent usage of the tool) and we can also use a simple thesaurus that relates main keywords with its synonymous. In fact, with time, the tool helps to build an ad-hoc thesaurus for each website.

3.3 Supplementary Tasks

Our Web mining model also performs mining of *frequent query patterns*, *text clustering* and *structure analysis* to complete the information provided by different query classes. We will present a brief overview of these tasks.

Frequent Query Patterns. All of the user queries are analyzed to discover frequent item sets (or frequent query patterns). Every keyword in a query is considered as an item. The discovered patterns contribute general information about the most frequent word sets used in queries. The patterns are then compared to the number of results given in each case by the internal search engine, to indicate if they are answered in the website or not. If the most frequent patterns don't have answers in the website, then it is necessary to review these topics to improve these contents more in depth.

Text Clustering. Our mining model clusters the website's documents according to their text similarity (the number of clusters is a parameter to the model). This is done to obtain a simple and global view of the distribution of content amongst documents, viewed as connected components in clusters, and to compare this to the website link organization. This feature is used to find documents with similar text that don't have links between them and that should be linked to improve the structure in the website. This process generates a visual report, that allows the webmaster of the website to evaluate the suggested improvements. At this point, it is important to emphasize that we are not implying that all of the documents with similar text should be linked, nor that this is the only criteria to associate documents, but we consider this a useful tool to evaluate in a simple, yet helpful way, the interconnectivity in websites (specially large ones).

The model additionally correlates the clustering results with the information about query classification. This allows to learn which documents inside each cluster belong to AD and BD sets and the frequency with which these events occur. This supports the idea of adding new groups of documents (topics) of interest to the top level distribution of contents of the website and possibly focusing the website to the most visited clusters, and also gives information on how documents are reached (only browsing or searching).

4 Evaluation

To test our model we used our prototype on several websites that had an internal search engine, the details of the prototype can be found in [26]. We will present some results from two of those sites: the first one, the website of a company dedicated to providing domain name registrations and services, and the second one, a portal targeted at university students and future applicants.

First Use Case. In Table 2 we present some results from the different query classes obtained for the fist use case. This site does not have a large amount of documents (approximately 1,130 documents) and its content, rather technical, seems quite straightforward. We believe this was the reason for finding only class A, B, C, D and E queries, but no class C' queries in its reports.

In Table 2 we have several suggestions for additional IS obtained from class A queries. Class B queries shown in this sample are very interesting, since they indicate which terms provide new IS for anchor text of documents about "nslookup", "CIDR", "trademarks" and "Web domains", which were topics not found by

Table 2. Sample of class A, B, C and D queries for the first use case

Class A	Class B	Class C	Class D
domains	nslookup	hosting	ASN
Internet providers	CIDR	DNS	
syntax	trademarks	server	
electronic invoice	lottery	prices	
diagnosis tools	Web domain	web hosting	

browsing in the site. Another interesting query in class B is "lottery", which shows a current popular topic within the registered domains in the site and makes a good suggestion for a hotlink in the top pages of the website. On the other hand, class C queries show that documents related mainly to topics on "Web hosting services" should be developed more in depth in the website. The only class D query found for this site, was "ASN", which stands for a unique number assigned by the InterNIC that identifies an autonomous system in the Internet. This is a new topic that was not present in the contents of the site at the moment of our study.

Second Use Case. The second use case, the portal targeted at university students and future applicants, was the primary site used for our evaluation in this paper. This site has approximately 8,000 documents, 310,000 sessions, 130,000 external and 14,000 internal queries per month. Using our model reports were generated for four months, two months apart from each other. The first two reports were used to evaluate the website without any changes, and show very similar results amongst each other. For the following reports, improvements suggested from the evaluation were incorporated to the site's content and structure. In this approach, the 20 most significant suggestions from the particular areas of: "university admission test" and "new student application", were used. This was done to target an important area in the site and measure the impact of the model's suggestions. A sample of frequent query patterns found in the website is shown in Table 3 and a sample of class A, B, C, C' and D queries is presented in Table 4.

The improvements were made mainly to the top pages of the site, and included adding IS to link descriptions, adding new relevant links, suggestions extracted from frequent query patterns, class A and B queries. Other improvements consisted of broadening the contents on certain topics using class C queries, and adding new contents to the site using class D queries. For example the site was improved to include more admission test examples, admission test scores and more detailed information on scholarships, because these where issues constantly showing in class C and D queries. To illustrate our results we will show a comparison between the second and third report. Figures 3, 4 and 5 show the changes in the website after applying the suggestions. For Figure 5 the queries studied are only the ones that were used for improvements.

In Figure 3 we present the variation in the general statistics of the site. After the improvements were made, an important increase in the amount traffic from

Table 3. Sample of frequent query patterns for the second use case (indicating which ones had few answers)

Percent(%)	Frequent pattern
3.55	admission test results
2.33	admission test scores
1.26	application results
1.14	scholarships
1.10	tuition fees
1.05	private universities
0.86	institutes
0.84	law school
0.80	career
0.74	courses
0.64	admission score
0.61	student loan
0.58	admission score
0.57	nursing
0.55	practice test *(only 2 results)*
0.54	engineering
0.53	psychology
0.53	credit
0.51	registration
0.51	grades
0.51	admission results *(only 2 results)*
0.49	architecture
0.44	student bus pass *(only one answer)*

Table 4. Sample of class A, B, C, C' and D queries for the second use case

Class A	Class B
practice test	university scholarships
thesis	admission test
admission test preparation	admission test inscription
university ranking	curriculum vitae
private universities	presentation letter
employment	bookstores

Class C	Class C'	Class D
admission test	government scholarships	Spain scholarships
admission test results	diploma	waiting lists
practice test	evening school	vocational test
scholarships	mobility scholarship	compute test score
careers	humanities studies	salary

external search engines is observed (more than 30% in two months), which contributes to an increase in the average number of page views per session per day, and also in the number of sessions per day. The increase in visits from external search engines is due to the improvements in the contents and link descriptions

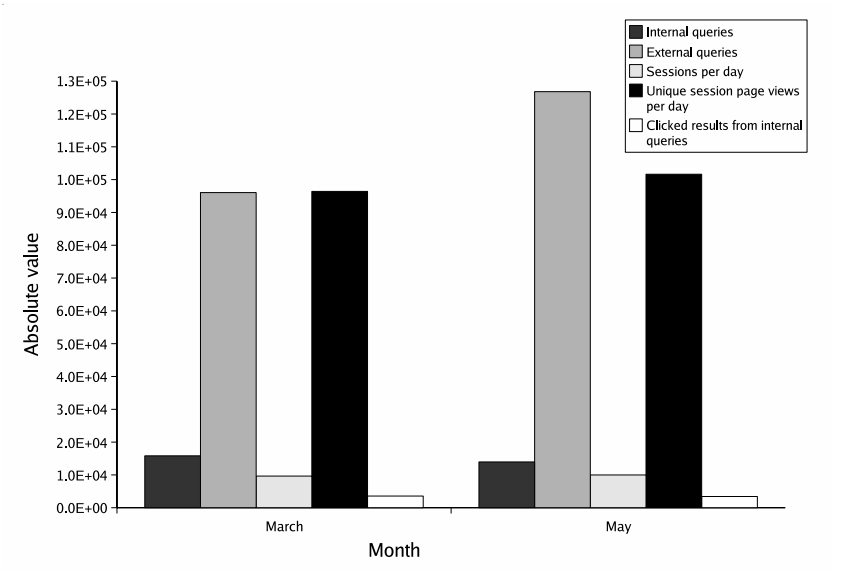


Fig. 3. General results

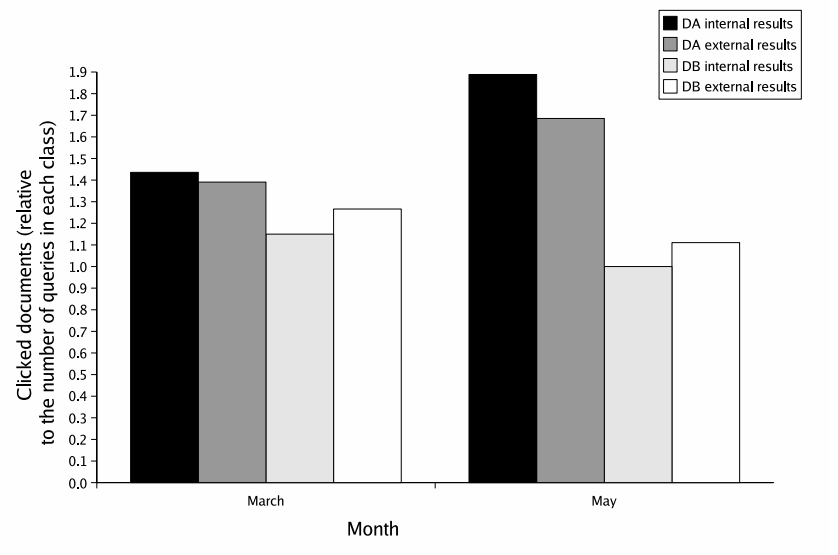


Fig. 4. Clicked results

in the website, validated by the keywords used on external queries. After the improvements were made to the site, we can appreciate a slight decrease in the number of internal queries and clicked documents from those queries. This agrees with our theory that contents are being found more easily in the website and that now less documents are accessible only through the internal search engine. All of these improvements continue to show in the next months of analysis.

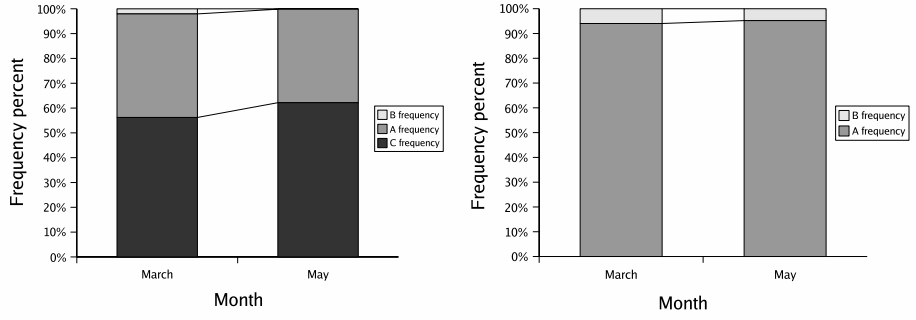


Fig. 5. Internal *(left)* and external *(right)* query frequency

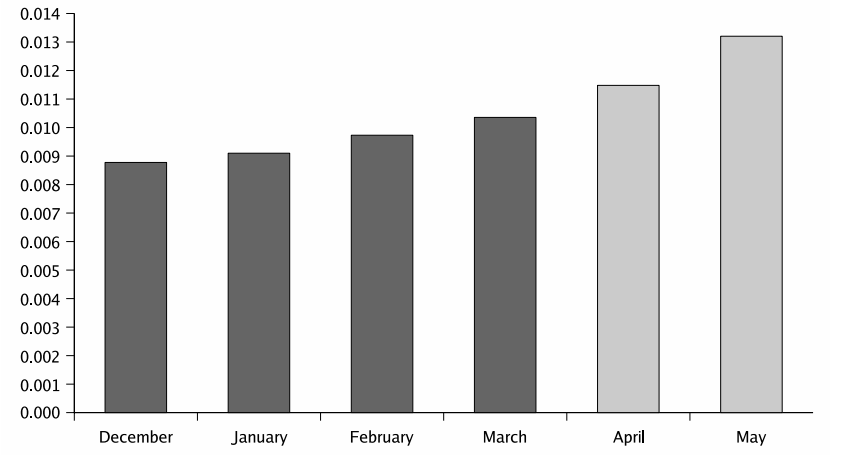


Fig. 6. Daily average number of external queries per month (normalized by the number of sessions)

Figure 4 shows the comparison between the number of documents (results) clicked from each query class, this number is relative to the numbers of queries in each class. External and internal AD documents present an important increase, showing that more external queries are reaching documents in the website, and that those documents now belong to documents that are being increasingly reached by browsing also. On the other hand BD documents continue to decrease in every report, validating the hypothesis that the suggested improvements cause less documents to be only reached by searching. In Figure 5 the distribution of A, B and C queries can be appreciated for internal and external queries. Internal queries show a decrease in the proportion of A and B queries, and an increase in queries class C. For external queries, class A queries have increased and class B queries have decreased, as external queries have become more directed at AD documents.

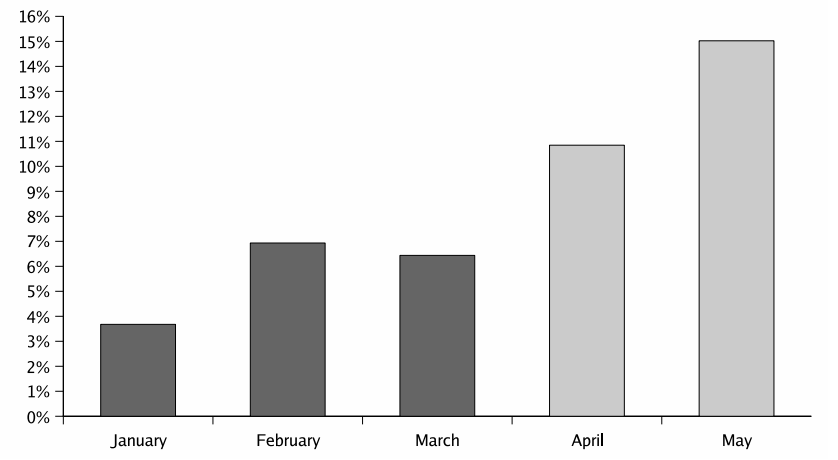


Fig. 7. Month to month percent variation of the daily average number of external queries (normalized by the number of sessions)

Figures 6 and 7 show statistics related to the amount of external queries in the website in months previous to the application of the model's suggestions and for the two months during and after they were applied (April and May). Usage data for the month of February was incomplete in Figure 6 (due to circumstances external to the authors) and had to be generated using linear interpolation with the months unaffected by our study. The data presented in Figures 6 and 7 show a clear increase above average in the volume of external queries that reached the website during April and May, specially in the month of May when the increase was in 15% compared to April, which is coherent with the fact that the results from the prototype where applied at the end of March.

5 Conclusions and Future Work

In this paper we presented the first website mining model that is focused on query classification. The aim of this model is to find better IS, contents and link structure for a website. Our tool discovers, in a very simple and straight forward way, interesting information. For example, class D queries may represent relevant missing topics, products or services in a website. Even if the classification phase can be a drawback at the beginning, in our experience, on the long run it is almost insignificant, as new frequent queries rarely appear. The analysis performed by our model is done offline, and does not interfere with website personalization. The negative impact is very low, as it does not make drastic changes to the website. Another advantage is that our model can be applied to almost any type of website, without significant previous requirements, and it can still generate suggestions if there is no internal search engine in the website.

The evaluation of our model shows that the variation in the usage of the website, after the incorporation of a sample of suggestions, is consistent with

the theory we have just presented. Even though these suggestions are a small sample, they have made a significant increase in the traffic of the website, which has become permanent in the next few reports. The most relevant results that are concluded from the evaluation are: *an important increase in traffic generated from external search engines*, *a decrease in internal queries*, *also more documents are reached by browsing and by external queries*. Therefore the site has become more findable in the Web and the targeted contents can be reached more easily by users.

Future work involves the development and application of different query ranking algorithms, improving the visualizations of the clustering analysis and extending our model to include the origin of internal queries (from which page the query was issued). Also, adding information from the classification and/or a thesaurus, as well as the anchor text of links, to improve the text clustering phase. Our work could also be improved in the future by analyzing query chains as discussed in [21] with the objective of using these sequences to classify unsuccessful queries, specifically class C' and E queries. Furthermore, we would like to change the clustering algorithm to automatically establish the appropriate number of clusters and do a deeper analysis of most visited clusters. The text clustering phase could possibly be extended to include stemming. Another feature our model will include is an incremental quantification of the evolution of a website and the different query classes. Finally, more evaluation is needed specially in the text clustering area.

References

1. Berendt, B., Spiliopoulou, M.: Analysis of navigation behaviour in web sites integrating multiple information systems. In: VLDB Journal, Vol. 9, No. 1 (special issue on "Databases and the Web"). (2000) 56–75
2. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N.: Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations **1**(2) (2000) 12–23
3. Cooley, R., Tan, P.N., Srivastava, J.: Discovery of interesting usage patterns from web data. In: WEBKDD. (1999) 163–182
4. Baeza-Yates, R.: Web usage mining in search engines. In: Web Mining: Applications and Techniques, Anthony Scime, editor. Idea Group (2004) 307–321
5. Perkowitz, M., Etzioni, O.: Adaptive web sites: an AI challenge. In: IJCAI (1). (1997) 16–23
6. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on web usage mining. Commun. ACM **43**(8) (2000) 142–151
7. Spiliopoulou, M.: Web usage mining for web site evaluation. Commun. ACM **43**(8) (2000) 127–134
8. Batista, P., Silva, M.J.: Mining on-line newspaper web access logs. In Ricci, F., Smyth, B., eds.: Proceedings of the AH'2002 Workshop on Recommendation and Personalization in eCommerce. (2002) 100–108
9. Cooley, R., Tan, P., Srivastava, J.: Websift: the web site information filter system. In: KDD Workshop on Web Mining, San Diego, CA. Springer-Verlag, in press. (1999)

10. Masseglia, F., Poncelet, P., Teisseire, M.: Using data mining techniques on web access logs to dynamically improve hypertext structure. ACM SigWeb Letters vol. 8, num. 3 (1999) 1–19
11. Huang, Z., Ng, J., Cheung, D., Ng, M., Ching, W.: A cube model for web access sessions and cluster analysis. In: Proc. of WEBKDD 2001. (2001) 47–57
12. Nasraoui, O., Krishnapuram, R.: An evolutionary approach to mining robust multi-resolution web profiles and context sensitive url associations. Intl' Journal of Computational Intelligence and Applications, Vol. 2, No. 3 (2002) 339–348
13. Nasraoui, O., Petenes, C.: Combining web usage mining and fuzzy inference for website personalization. In: Proceedings of the WebKDD workshop. (2003) 37–46
14. Pei, J., Han, J., Mortazavi-asl, B., Zhu, H.: Mining access patterns efficiently from web logs. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. (2000) 396–407
15. Perkowitz, M., Etzioni, O.: Adaptive web sites: automatically synthesizing web pages. In: AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, Menlo Park, CA, USA, American Association for Artificial Intelligence (1998) 727–732
16. Xue, G.R., Zeng, H.J., Chen, Z., Ma, W.Y., Lu, C.J.: Log mining to improve the performance of site search. In: WISEW '02: Proceedings of the Third International Conference on Web Information Systems Engineering (Workshops) - (WISEw'02), Washington, DC, USA, IEEE Computer Society (2002) 238
17. Baeza-Yates, R.A., Hurtado, C.A., Mendoza, M.: Query clustering for boosting web page ranking. In Favela, J., Ruiz, E.M., Chávez, E., eds.: AWIC. Volume 3034 of Lecture Notes in Computer Science., Springer (2004) 164–175
18. Baeza-Yates, R.A., Hurtado, C.A., Mendoza, M.: Query recommendation using query logs in search engines. In Lindner, W., Mesiti, M., Türker, C., Tzitzikas, Y., Vakali, A., eds.: EDBT Workshops. Volume 3268 of Lecture Notes in Computer Science., Springer (2004) 588–596
19. Kang, I.H., Kim, G.: Query type classification for web document retrieval. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM Press (2003) 64–71
20. Sieg, A., Mobasher, B., Lytinen, S., Burke, R.: Using concept hierarchies to enhance user queries in web-based information retrieval. In: IASTED International Conference on Artificial Intelligence and Applications. (2004)
21. Radlinski, F., Joachims, T.: Query chains: learning to rank from implicit feedback. In: KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, New York, NY, USA, ACM Press (2005) 239–248
22. Davison, B.D., Deschenes, D.G., Lewanda, D.B.: Finding relevant website queries. In: Poster Proceedings of the Twelfth International World Wide Web Conference, Budapest, Hungary (2003)
23. Baeza-Yates, R.: Mining the web (in spanish). El profesional de la información (The Information Professional) **13**(1) (2004) 4–10
24. Cooley, R., Mobasher, B., Srivastava, J.: Data preparation for mining world wide web browsing patterns. Knowledge and Information Systems **1**(1) (1999) 5–32
25. Mobasher, B.: Web usage mining and personalization. In Singh, M.P., ed.: Practical Handbook of Internet Computing. Chapman Hall & CRC Press, Baton Rouge (2004)

26. Poblete, B.: A web mining model and tool centered in queries. M.sc. in Computer Science, CS Dept., Univ. of Chile (2004)
27. Pirolli, P.: Computational models of information scent-following in a very large browsable text collection. In: CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM Press (1997) 3–10

Author Index