DIR 2009

Dutch-Belgian Information Retrieval Workshop

PROCEEDINGS OF THE 9TH DUTCH-BELGIAN
INFORMATION RETRIEVAL WORKSHOP (DIR 2009)

FEBRUARY 2-3, 2009
ENSCHEDE, THE NETHERLANDS

Robin Aly, Claudia Hauff, Ida den Hamer,
Djoerd Hiemstra, Theo Huibers, Franciska de Jong (eds)

Book orders:
Secretariat CTIT
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands
Phone +31 53 489 80 31

Druk- en bindwerk: Ipskamp Drukkers, Enschede, The Netherlands

CIP GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Aly, R.B.N., Hauff, C., den Hamer-Mulder, I.C.,
Hiemstra, D., Huibers, T.W.C, de Jong, F.M.G.

Proceedings of the 9th Dutch-Belgian Information Retrieval Workshop
CTIT Workshop Proceedings WP 09-01
Enschede, University of Twente, Centre for Telematics and Information Technology
ISSN 0929-0672; No 09-01 (CTIT Workshop Proceedings Series)
Subject headings: information retrieval, search engines, databases

Preface

Welcome to the 9th Dutch-Belgian Information Retrieval Workshop (DIR). I very well remember the DIR workshop in 2001 that was also organized in Twente. It took place exactly one day before my PhD defense, to give us the opportunity to have one of the PhD committee members, Stephen Robertson, as the keynote speaker. I am proud to see that DIR does not need PhD defenses any more to attract excellent keynotes. This year, DIR presents Rene van Erk, Director of Product- and Business Development Europe at Wolters Kluwer as the *industry keynote*; we present professor Gerhard Weikum, Scientific Director at the Max-Planck Institute for Informatics, as the *academic keynote*. We also present comedian Daniel van Veen as the "*cultural keynote"* to let our participants taste a bit of our university's unique campus activities.

We have tried this year to especially encourage PhD students and researchers from industry to submit their research. Each submission was reviewed by at least two and often three or four program committee members. We thank the program committee members for their high quality reviews. Of 15 submissions to DIR, 12 were accepted. Another 5 submissions were accepted as poster presentations. Four papers were written with participation from industry and most other papers have a PhD student as the first author.

Special thanks to the Netherlands Research School for Information and Knowledge Systems (SIKS) for sponsoring the participation of Dutch SIKS members, to the Werkgemeenschap Informatiewetenschap (WGI) for providing a solid financial basis for organizing DIR now and in the future, to the Netherlands Organization of Scientific Research (NWO) for sponsoring the travel and hotel costs of our international keynote speaker Gerhard Weikum, to the Centre for Telematics and Information Technology (CTIT) for sponsoring the proceedings, to the University of Twente and to Cultural Center the Vrijhof for sponsoring our cultural activity on Monday.

To an inspiring workshop!

Djoerd Hiemstra.

Organizing committee

- Robin Aly
- Claudia Hauff
- Ida den Hamer-Mulder
- Djoerd Hiemstra
- Theo Huibers
- Franciska de Jong

Program committee

- Leif Azzopardi (University of Glasgow, UK)
- Antal van den Bosch (Tilburg University, Netherlands)
- Gosse Bouma (University of Groningen, Netherlands)
- Martine de Cock (Ghent University, Belgium)
- Walter Daelemans (University of Antwerp, Belgium)
- Anne Diekema (Utah State University, USA)
- Leo Egghe (University of Antwerp, Belgium)
- Theo Gevers (University of Amsterdam, Netherlands)
- Eduard Hoenkamp (University of Maastricht, Netherlands)
- Veronique Hoste (Hogeschool Gent, Belgium)
- Jaap Kamps (University of Amsterdam, Netherlands)
- Wessel Kraaij (Radboud University Nijmegen and TNO-ICT, Netherlands)
- Marie-Francine Moens (University of Leuven, Belgium)
- Stephan Raaijmakers (TNO-ICT, Netherlands)
- Maarten de Rijke (University of Amsterdam, Netherlands)
- Henning Rode (CWI Amsterdam, Netherlands)
- Hichem Sahli (Vrije Universiteit Brussel, Belgium)
- Guy De Tre (Ghent University, Belgium)
- Tinne Tuytelaars (University of Leuven, Belgium)
- Remco Veltkamp (Utrecht University, Netherlands)
- Werner Verhelst (Vrije Universiteit Brussel, Belgium)
- Paul van der Vet (Universiteit Twente, Netherlands)
- Arjen de Vries (Delft University of Technology and CWI Amsterdam, Netherlands)
- Jun Wang (University College London, UK)
- Theo van der Weide (Radboud University Nijmegen, Netherlands)
- Thijs Westerveld (Teezir Search Solutions, Netherlands)
- Roelof van Zwol (Yahoo!, Barcelona, Spain)

Sponsors

Netherlands Research School for Information and Knowledge Systems
http://www.siks.nl

Vereniging Werkgemeenschap Informatiewetenschap
http://www.informatiewetenschap.org

Netherlands Organisation for Scientific Research
http://www.nwo.nl

Centre for Telematics and Information Technology, SRO NICE
http://www.ctit.utwente.nl

University of Twente
http://www.utwente.nl

Vrijhof Cultuurcentrum
Kunst en cultuur op de Universiteit Twente
http://www.cultuur.utwente.nl

Table of contents

Keynote addresses

Paper session: Multimedia & XML

Paper session: Question answering & query formulation

Paper session: Content modeling & keyword extraction

Paper session: Spam detection, topic modeling & information extraction

Poster & demo session

Program

Monday February 2nd, 2009

10.30	-	11.00	Welcome, registration, and coffee
11.00	-	12.00	**Session 1: Industry keynote (Audiozaal) session chair: Theo Huibers** **Communities and Workflow: Driving Information Consumption** Speaker: Rene van Erk (*Wolters-Kluwer, Netherlands*)
12.00	-	12.35	Lunch (lunch can be taken to Session 2)
12.35	-	13.45	**Session 2: Cultural keynote "Broodje Cultuur" (Amphitheater)** **Van Veen Alleen (Van Veen Alone)** Speaker: Daniël van Veen
13.45	-	15.15	**Session 3: Multimedia & XML (Audiozaal), session chair: Arjen de Vries**
			Automatic Extraction of Keywords for a Multimedia Search Engine Using the Chi-Square Test Marco A. Palomino, Michael P. Oakes (*University of Sunderland, UK)* and Tom Wuytack (*Belga News Agency, Belgium*)
			Spatiotemporal Region-oriented Attentional Selection of Active Salient Objects Thomas Geerinck, Hichem Sahli, Iris Vanhamel and Valentin Enescu (*Vrije Universiteit Brussel, Belgium*)
			Exemelification of Parliamentary Debates Tim Gielissen and Maarten Marx (*University of Amsterdam, Netherlands*)
15.15		15.45	Coffee/tea
15.45	-	17.15	**Session 4: Question answering & query formulation (Audiozaal), session chair: to be announced**
			Matching Queries to Frequently Asked Questions: Search Functionality for the MRSA Web-Portal Almer S. Tigelaar, Rieks op den Akker and Fenne Verhoeven (*University of Twente, Netherlands*)
			Learning to Rank Answers to Why-Questions Suzan Verberne, Daphne Theijssen, Lou Boves (*Radboud University Nijmegen, Netherlands*) and Stephan Raaijmakers (*TNO Delft, Netherlands*)
			Non-Linear Query Reformulation Behavior Nikolai Buzikashvili (*Russian Academy of Science, Russia*)
17.15	-	18.15	**Session 5: Poster & demo session**
18.30	-	20.30	Dinner at the Faculty Club

Tuesday February 3rd, 2009

09.00	-	10.00	**Session 6: Academic keynote (Audiozaal), session chair: Djoerd Hiemstra** **Harvesting, Searching, and Ranking Knowledge on the Web** Speaker: Prof. Gerhard Weikum (*Max-Planck Institute for Informatics, Saarbrücken, Germany*)
10.00	-	10.30	Coffee/tea
10.30	-	12.00	**Session 7: Content modeling & keyword extraction (Audiozaal), session chair: Eduard Hoenkamp** **Language Independent Context Extraction from Web Pages** Javier Arias Moreno, Koen Deschacht and Marie-Francine Moens (*KU Leuven, Belgium*) **Patent Search: An important new Test Bed for IR** John Tait, Mihai Lupu, (*Information Retrieval Facility, Austria*), Helmut Berger, Giovanna Roda, Michael Dittenbach, Andreas Pesenhofer (*Matrixware, Austria*), Erik Graf and Keith van Rijsbergen (*University of Glasgow, UK*) **Semi-Automatic Construction of Domain-Specific Thesauri** Inga Kohlhof, Bob Schijvenaars (*SyynX Solutions GmbH, Germany*) and Mario Diwersy (*Collexis Holdings, USA*)
12.00	-	13.00	Lunch
13.00	-	14.30	**Session 8: Spam detection, topic modeling & information extraction, session chair: Marie-Francine Moens** **Web Directories as Topical Context** Rianne Kaptein and Jaap Kamps (*University of Amsterdam, Netherlands*) **Business Specific Online Information Extraction from German Websites** Yeong Su Lee and Michaela Geierhos (*University of Munich, Germany*) **Using Language Modeling for Spam Detection in Social Reference Manager Websites** Toine Bogers and Antal van den Bosch (*Tilburg University, Netherlands*)
14.30	-	15.00	Closing
15.00	-	16.30	Annual Meeting of the Werkgemeenschap Informatiewetenschap

Communities and Workflow: Driving Information Consumption

Keynote presentation

Rene van Erk
Wolters-Kluwer, The Netherlands
rene.van.erk@wolterskluwer.com

Abstract

A good understanding of communities and knowledge on workflows are key drivers for product innovation. The dynamics of the "old" publishing world need to be brought together with the dynamics of the new information world, characterized by 2.0 solutions, social media, gadgets, etc.

A vision on communities. For any product or service provider it is important to have a vision on communities: the internet is creating new information usage patterns which drive opportunities for user generated content, social networking, and changing revenue streams. Communities are key, but they can take time to build and a community manager to facilitate dialogue can help to add value.

Insight into workflows. To drive innovative products and solutions, it is also important to have an insight into workflow needs. Customer interaction and testing show that workflow solutions cannot be stand-alone anymore but must provide integrated content, and add-on features must be accessible from within the same environment. Additionally, different customers have different priorities and need different user-interfaces. A doctor's needs are very different from those of an accountant, for example. It is essential to know the dynamics of different customer groups.

LA LEY Digital. Wolters Kluwer Spain's LA LEY digital builds on the recognition that individuals who are part of a community are more motivated to provide feedback then individuals who are not. The product incorporates a solid understanding of customers' daily priorities and work. LA LEY features innovative functionalities such as:

- Semantic Expansion (including synonyms in the search as well as recognizing concepts rather than just words)
- "Did You Mean" (recognizing typos in a search query)
- Dynamic clustering – Terms Suggestion
- Faceted queries – Metadata clustering (users can filter results by all the metadata on the result list)
- Key Word in Context
- Dynamic summary

Some of these features, such as the Semantic Expansion as well as the Dynamic Summary, are innovative features which are rarely seen in any product worldwide. The feedback on the product so far is encouraging. The number of user-sessions has increased by 40 % in the last three months and the time per session has also increased by about 40 %.

Bio

Rene is currently part of the European management team of Wolters Kluwer where he is responsible for all Product âĂŞ and Business Development. In this role, his key responsibility is to optimize the WK portfolio for maximum growth, meaning:

- M&A focus: Responsible for identifying acquisition opportunities with a good strategic portfolio
- Leading Innovation: Overall responsible for Product Development & Product Management moving WK from content provider to information solutions provider.
- Leading our Online and Software Businesses: WK currently owns around 30 Software Development Companies and leading Online properties across Europe

Prior to Wolters Kluwer, but after having returned to the Netherlands in 2004, Rene was an informal investor in a number of Internet and Software start-ups; he is still a member of the Board in 1 of those. He was also part of the management team that led the turnaround of Tridion, a leading European Content Management- and Online Solutions provider.

Between 1991 and 2004 Rene worked in a number of Software companies, but spend most of his time -âĂŞ 9 years – in various roles across the globe for Microsoft. His last role in Microsoft was to lead the Asia-Pacific Services & Solutions business (1700 FTE) through a transition towards a vertical (vs geographical) business. As part of the Telecom Solutions practice for example (to drive triple play projects), he managed a/o Microsoft's investment of 500M$ in Korea Telecom implementing IPTV, Video on Demand, B2B Marketplace etc etc. In that position he also established a Global Solution Development Center in Bangalore (India) and started a global support center in Hyderabad (1500 FTE today). Before Microsoft Rene worked in two other software companies, establishing/growing international businesses for both of them.

Rene prefers operating in an international environment, speaks 5 languages and has worked across multiple continents; he lived in/worked in Asia-Pacific, North America and Europe. He is passionate about Internet, Mobility, Digital Media/Content and the convergence with telecommunication.

In his spare time he enjoys cooking, good wine and traveling, as well as watching his two sons (6 and 7) play soccer. He holds a degree in Business Administration.

Harvesting, Searching, and Ranking Knowledge on the Web

Keynote presentation

Gerhard Weikum
Max Planck Institute for Informatics
Saarbruecken, Germany
weikum@mpi-inf.mpg.de

Abstract

There are major trends to advance the functionality of search engines to a more expressive semantic level. This is enabled by employing large-scale information extraction of entities and relationships from semistructured as well as natural-language Web sources. In addition, harnessing Semantic-Web-style ontologies and reaching into Deep-Web sources can contribute towards a grand vision of turning the Web into a comprehensive knowledge base that can be efficiently searched with high precision.

This talk presents ongoing research towards this objective, with emphasis on our work on the YAGO knowledge base and the NAGA search engine but also covering related projects. YAGO is a large collection of entities and relational facts that are harvested from Wikipedia and WordNet with high accuracy and reconciled into a consistent RDF-style "semantic" graph. For further growing YAGO from Web sources while retaining its high quality, pattern-based extraction is combined with logic-based consistency checking in a unified framework. NAGA provides graph-template-based search over this data, with powerful ranking capabilities based on a statistical language model for graphs. Advanced queries and the need for ranking approximate matches pose efficiency and scalability challenges that are addressed by algorithmic and indexing techniques.

YAGO is publicly available and has been imported into various other knowledge-management projects including DBpedia. YAGO shares many of its goals and methodologies with parallel projects along related lines. These include Avatar, Cimple/DBlife, DBpedia, KnowItAll/TextRunner, Kylin/KOG, the Libra technology, and more. Together they form an exciting trend towards providing comprehensive knowledge bases with semantic search capabilities.

This is joint work with Shady Elbassuoni, Georgiana Ifrim, Gjergji Kasneci, Thomas Neumann, Maya Ramanath, Mauro Sozio, and Fabian Suchanek.

Bio

Gerhard Weikum is a Scientific Director at the Max-Planck Institute for Informatics, where he is leading the research group on databases and information systems. Earlier he held positions at Saarland University in Germany, ETH Zurich in Switzerland, MCC in Austin, and he was a visiting senior researcher at Microsoft Research in Redmond. His recent working areas include peer-to-peer information systems, the integration of database-systems and information-retrieval methods, and information extraction for building and maintaining large-scale knowledge bases. Weikum has co-authored more than 150 publications, including a comprehensive textbook on transactional concurrency control and recovery. He received several best paper awards including the VLDB 2002 ten-year award, and he is an ACM Fellow. He has served on the editorial boards of various journals and book series, including ACM TODS, the Springer LNCS series, and the new CACM, and as program committee chair for international conferences like ICDE 2000, ACM SIGMOD 2004, and CIDR 2007. He is currently the president of the VLDB Endowment.

Automatic Extraction of Keywords for a Multimedia Search Engine Using the Chi-Square Test

Marco A. Palomino
University of Sunderland
Informatics Centre
United Kingdom
marco.palomino@sund.ac.uk

Michael P. Oakes
University of Sunderland
Informatics Centre
United Kingdom
michael.oakes@sund.ac.uk

Tom Wuytack
Belga News Agency
1030 Brussels
Belgium
wut@belga.be

ABSTRACT

We present a method to automatically acquire a set of keywords that characterise a large multimedia collection. Our method compares captions associated with pictures in the collection with a model of general English language. The words that deviate from the model are very specific of the captions and thus make appropriate keywords. Professional annotators evaluated our results and concluded that more than 97% of our top 2,000 one-word keywords were truly descriptive of the collection. We also mined the collection's query logs and extracted keywords that reflect the most important indexing terms from the users' perspective. Our method offers a strategy for selecting the keywords that make up the indices of multimedia search engines.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing.

General Terms

Algorithms.

Keywords

Chi-square test, indexing terms, search logs, collocation.

1. INTRODUCTION

The fundamental operation performed by a search engine is a *match*, between descriptive features mentioned by the users in their queries and the documents sharing those features. By far, the most important kind of features are *keywords*, linguistic atoms—typically words, pieces of words, or phrases—used to characterise the subject or content of a document [1]. Keywords are pivotal because they must bridge the gap between the users' characterisation of information need—i.e., their queries—and the characterisation of the documents' topical focus against which these queries will be matched.

DIR 2009 University of Twente, The Netherlands

Indexing is the process by which a vocabulary of keywords is assigned to all the documents of a corpus [1]. Such an assignment can be done manually or automatically. *Manual indexing* requires people with expertise in the domain of discourse, who should read each document and select appropriate keywords for it. *Automatic indexing* refers to algorithmic procedures for accomplishing this same result. Since the index relation is the fundamental connection between the user queries and the documents that can satisfy them, choosing a suitable vocabulary of keywords is at the core of the information retrieval problem.

As part of the EU-funded *VITALAS* project [27], which aims to provide a prototype system dedicated to the intelligent access to professional multimedia archives, we have experimented with several techniques for the automatic extraction of keywords. Our goal is to obtain keywords that typify the content of a large multimedia collection owned by the *Belga News Agency* [2]. Belga's collection covers national and international news, and all aspects of life and current affairs—politics and economics, finance and social affairs, sports, culture and personalities. Although Belga's content is published in four different formats—text, pictures, audio and video—, this paper concentrates on pictures and their associated textual captions, exclusively.

Due to the immense size of Belga's collection—more than 300 pictures per day—we cannot afford the choice of indexing the entire collection manually. Thus, we have selected an automatic approach. The method documented here for the automatic extraction of keywords is based on the *chi-square test*, a well-known statistical technique described by various authors [4, 17]. The results yielded by the chi-square test were evaluated by professional annotators, who confirmed whether the keywords that we retrieved were descriptive of the collection and likely to make suitable indexing terms. Most of the keywords that were rejected were those made of names and initials of photographers and acronyms of press agencies, which we could not eliminate from our analysis for reasons that we discuss later.

The remainder of this paper is organised as follows: Section 2 offers an overview of related work; Section 3 explains the rationale behind our use of the chi-square test; Section 4 describes our lexical analysis of Belga captions; Section 5 justifies our choice of a comparison corpus; Section 6 reports on the computation of the chi-square test; Section 7 presents the annotators' feedback; Section 8 reports on our performance analysis and Section 9 explores the use of query logs as an alternative source for keyword extraction. Lastly, Section 10 states our conclusions.

2. RELATED WORK

Some of the most successful algorithms to extract keywords automatically are supervised learning algorithms. Hulth and Megyesi [6], for instance, devised an algorithm that employs three prediction models trained on manually annotated data and combines them to extract keywords from scientific journal abstracts.

While we are aware of the advantages of training machine-learning systems, we are discouraged by their inability to adapt to new languages and domains easily, as additional training data is required for each new collection. We have therefore chosen a different approach.

Our methodology to extract keywords bears certain resemblance to the one employed by the *OmniPaper* system [13] to discover suitable terms to index newspaper articles. As in the case of OmniPaper, we implement a tokenisation process that includes stop-word filtering, stemming and proper-name detection. Nevertheless, OmniPaper selects keywords according to document frequency and a choice of frequency thresholds that tune up its performance. Our approach, in contrast, is based on the application of the chi-square test. Also, it is worth noting that OmniPaper's keywords were not reviewed by professional annotators who knew the collection and assessed their appropriateness.

The use of the chi-square test to find words that are truly characteristic of a corpus has been documented before. Leech and Fallon used the chi-square test to uncover the differences in the vocabulary of the *Brown Corpus*—one million words of American English—and the *Lancaster-Oslo-Bergen Corpus*—one million words of British English—in fifteen categories, such as sport, transport and travel, and business [12]. Rayson *et al.* used the same test with the *British National Corpus* to study the variations between the vocabulary used by male and female speakers [19], and, recently, Oakes and Farrow used it to examine vocabulary differences in English language corpora representing seven different countries [15].

Our interest in *collocations*—i.e., sequences of words that co-occur more often than would be expected by chance—which is addressed in Subsection 6.2, is motivated by an increasing concern in the NLP community to deal with multi-word expressions, regardless of language and domain [21]. Indeed, according to Jackendoff [9] and Mel'čuk [14], collocations constitute the bulk of a language's lexicon.

In the past few years, considerable development in collocation extraction techniques has taken place. The main heuristics in this field have been described by Seretan and Wehrli [22]. Certainly, we plan to use one of those multilingual heuristics in the near future, but, at present, our strategy is restricted to co-occurrence frequency—i.e., the systematic occurrence of two lexical items within a fixed distance in the corpus—as proposed by Smadja [25].

3. THE CHI-SQUARE TEST

The chi-square test is a useful statistical technique to decide if observed data deviates from expected data under a particular hypothesis [4]. Such a hypothesis is whether two samples are different enough in a specific characteristic to be considered members of different populations. In this sense, the chi-square test is used to assess the *null hypothesis* (H_0), which states that there is no significant difference between observed and expected data.

Chi-square values with low probability lead to the rejection of H_0, and it is assumed that a factor, other than chance, creates a deviation between expected and observed values. As with other non-parametric tests, the chi-square test does not take into account the interaction among more than one variable upon the outcome.

We have employed the chi-square test to extract keywords that are truly characteristic of the Belga multimedia collection. The chi-square test compared captions associated with pictures posted on the Belga website with a model of general English language. The words that deviated from the model were very specific of Belga data and thus made an appropriate set of keywords to search for Belga pictures and content on the Web.

4. BELGA CAPTIONS

To ensure that we had enough material to perform our research, the Belga News Agency allowed us to use a set of 1,727,159 captions that were published on its website between 22 June 2007 and 2 October 2007. A *caption* is a free-text field whose content is created by photographers, and offers an explanation or designation accompanying a picture posted on the Belga website.

Each group of photographers has its own conventions and styles to present the information. Hence, apart from text relative to pictures, certain captions may include names of photographers, their initials and acronyms of press agencies, as well as the dates when the pictures were taken or published. Since none of this information is deleted before its posting, we decided to keep it in our analysis too. Of course, this means that certain initials and acronyms—for instance, `EPA` (*European Pressphoto Agency*)—have a very high frequency of appearance.

We started our lexical analysis of Belga captions by carrying out a number of pre-processing steps commonly associated with the construction of an index,

Tokenisation: The process of splitting a piece of text into its parts, called *tokens*, while disposing of certain characters, such as punctuation and numbers. Tokens are loosely referred to as *words* in this paper.

Stop-word removal: The process of eliminating extremely common and semantically non-selective words. The stop-word list that we used was built by Salton and Buckley for the experimental *SMART information retrieval system* [3], and contains 571 words.

Stemming: The process of reducing inflectional forms and derivationally related forms of a word to a common base. The particular algorithm for stemming English words that we are using is *Porter's algorithm* [18].

5. COMPARISON CORPUS

The chi-square test requires a comparison between the corpus for which we want to find out its characteristic keywords and a second corpus, sometimes known as *comparison corpus*. Since we wanted to identify words that were truly characteristic of Belga captions, we required a comparison corpus that did not belong to the news-agency domain. Also, we needed a large corpus comparable in size to the total amount of Belga captions. After considering our choices, we opted for *Europarl* [11].

Europarl is a freely-available parallel corpus derived from the *Proceedings of the European Parliament*. It includes versions in 11 European languages: Romanic—French, Italian, Spanish, Portuguese—, Germanic—English, Dutch, German, Danish, Swedish—, Greek and Finnish. Europarl was mainly created to foster research in statistical machine translation, but it has been widely used by the computational linguistics community. It has been adopted in research related to word sense disambiguation, anaphora resolution and information extraction, among others.

We did consider the utilisation of other comparison corpora. Specifically, we contemplated the use of *Wikipedia* [28]. However, the availability of a "precise" translation of Europarl's corpus into other languages led us to its choice.

In the context of the VITALAS project, we aim to index other multimedia collections in the future, such as those owned by *Institut National de l'Audiovisuel* [7], which is in French, and *Institut für Rundfunktechnik* [8], which is in German. Hence, the possibility of using the same corpus in French and German presents a definite advantage for us.

Europarl supplies sentence aligned text for statistical machine translation systems, which means that we could use exactly the same corpus to analyse other collections, regardless of their language. Conversely, Wikipedia's corpus in languages other than English might be shorter than its English counterpart and not necessarily the same.

To the best of our knowledge, the use of Europarl does not impose any copyright restrictions; yet, some pre-processing is necessary before using its corpus. In particular, the removal of XML mark-up is indispensable. After tokenisation and removal of XML mark-up, Europarl's English-language corpus comprises 8,279,120 words.

6. CHI-SQUARE TEST COMPUTATION

For every distinct word that occurs in Belga captions, we need to find out how often it appears in Europarl and how often it occurs in Belga. For instance, the word *film*, which is found 40,466 times in Belga captions, occurs only 473 times in Europarl. We need to know whether these figures arose by chance, or whether the word *film* is genuinely more typical of Belga captions. Also, we need to know the total number of words in both corpora—Belga captions and Europarl—so that we can work out how many words are *not* the word *film*. All these values are arranged in a 2 by 2 table called a *contingency table*. Figure 1 shows the contingency table for the word *film*.

Number of times *film* is found in Belga	Number of times *film* is found in Europarl
Number of words other than *film* in Belga	Number of words other than *film* in Europarl

Figure 1: Contingency table for the word *film*

Since the total number of words and the number of appearances of a word in each corpus are values that we can count directly, they are called *observed values*. From the observed values, we can calculate the corresponding *expected values*, if there were no tendency for words to appear more often in either Belga captions or Europarl.

We calculate the chi-square value (χ^2) for a particular, word, whose contingency table is τ, as follows,

$$\chi^2 = \sum_{Each\ cell\ in\ \tau} \frac{(\varrho - \varepsilon)^2}{\varepsilon},$$

where ϱ is the observed value for each cell in τ, and ε is the expected value for each cell in τ. The expected value ε is calculated using the formula,

$$\varepsilon = \frac{Row\ total \cdot Column\ total}{Grand\ total}.$$

If the resulting chi-square value is more than 10.83, we can be 99.9% confident that the word really is more typical of Belga captions than of Europarl[1]. For example, since the chi-square value for *film* is $5,554.75$, we can be 99.9% confident that *film* is genuinely more typical of Belga captions than of Europarl.

6.1 Discussion

When comparing the frequencies of the vocabulary in two corpora, it is important to take into account only those words which are *well dispersed* across each corpus. In a comparison of the *FLOB corpus*—1990s British English—and the *Frown Corpus*—early 1990s American English—Oakes and Farrow [15] found that the 15^{th} most typical word of British English was "thalidomide", when considering the chi-square measure alone. However, all 55 occurrences of this word were found in a single article in the FLOB corpus. This problem was overcome by taking into account only those words for which Juilland's *D dispersion measure* [10] was above a suitable threshold. The problem of a single word occurring many times in a single caption did not arise in our study, because Belga captions are all short.

The *false discovery rate*, which is the inevitability of finding a number of spurious "significant" differences in word frequencies when performing a large number of comparisons—one for every word in the vocabulary—is another issue we were concerned about. This can be controlled by the *Bonferroni correction* [20].

We aimed to find words that were typical of Belga captions with 99% confidence, which corresponds to a 1% significance level. Applying the Bonferroni correction to the number of stems in the Belga vocabulary—249,960—gives us a new chi-square threshold value of 31.2137. All the words sent to the professional annotators for manual evaluation produced chi-square values greatly in excess of this number.

6.2 Collocations

A *collocation* is a sequence of words that co-occur more often than would be expected by chance. Finding multiple word combinations in a corpus for indexing purposes provides richer indexing units than single words [24, 25].

[1]A relative standard serves as the basis for accepting or rejecting H_0. In computational linguistics research, the relative standard is usually $q = 0.001$, where q is the probability that the deviation of the observed values from the expected values is due to chance alone. If q is less than or equal to 0.001, then H_0 is rejected, and the values are not independent from each other. The critical chi-square value associated with $q = 0.001$ is 10.83. Therefore, chi-square values higher than 10.83 are associated with a statistically low probability that H_0 is true.

Sometimes, a collocation gives the context in which a given word is used, which helps retrieve documents that use that word in the desired sense. For example, from a semantic point of view, the phrase *international film festival* is a better representation of a concept than the individual words *international*, *film* and *festival*.

First, we computed chi-square values for all the words contained in Belga captions. Then, we computed chi-square values for all the collocations made of two words and three words contained in Belga captions.

The only restriction when extracting collocations made of two words was that neither of the two words in the collocation could be stop-words. In the case of collocations made of three words, neither the first nor the last word could be stop-words—the middle one, however, could be a stop-word. Therefore, *January in Brussels* is a valid collocation, whereas *last January in* is not.

Figure 2 shows the architecture of our system for the automatic extraction of keywords. Text and metadata included in the captions is first submitted to our lexical analyser, which stems the words and remove the stop-words. Then, collocations made of one, two or three stems are extracted and prepared for the computation of the chi-square test. All the collocations are sorted by chi-square value in descending order, and those with a value above 31.2137 are stored as potentially useful keywords.

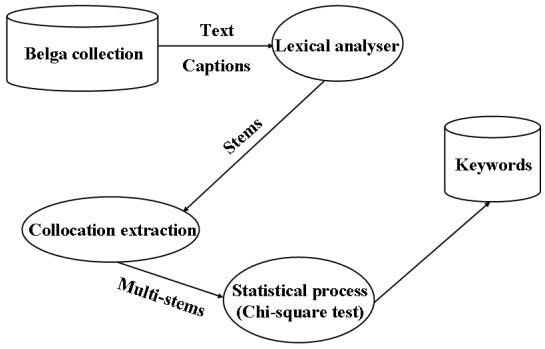


Figure 2: System architecture

7. RESULTS

Our results were divided into three categories: keywords made of one word—*singles*—, keywords made of two words—*pairs*—, and keywords made of three words—*triplets*. The annotators reviewed the 2,000 keywords with the highest chi-square values within each category. All these keywords had a chi-square value greater than 31.2137, which, as explained above, guaranteed that they were genuinely typical of Belga captions.

Professional annotators reviewed our keywords and decided whether they were descriptive of the collection. The review was performed in a binary fashion: each keyword was either accepted, which meant it was truly descriptive of the collection, or rejected, which meant it was not characteristic of the collection.

More than 97% of the singles derived from our analysis—i.e., 1,943 singles—were accepted by the annotators. Less than 3% of our singles—i.e., 58 singles—were considered unsuitable. Some of these unsuitable singles, such as `held` or `grand`, were rejected because they were too ambiguous to characterise Belga's picture collection.

The vast majority of unsuitable singles were those that refer to photographers' initials—for instance, `lg` (*Louisa Gouliamaki*) and `gw` (*Greg Wood*)—press agencies' acronyms—for example, `AFP` (*Agence France-Presse*), `ANP` (*Netherlands National News Agency*) and `VRT` (*Flemish Radio and Television Network*)—and days of the week and months of the year that dated the publication of a picture but did not describe its contents—such as `sundai` and `april`.

Approximately 70% of the pairs that we submitted for evaluation—i.e., 1,394 pairs—were accepted by the annotators too. Only 30% of our pairs—i.e., 605 pairs—were unsuitable. The unsuitable pairs were, mostly, those that contain names of photographers and acronyms of press agencies—for instance, `afp file`—combinations of publishers and dates when the pictures were taken—`februari epa` (a picture taken in February and published by the *European Pressphoto Agency*)—stop-words in languages other than English—`de la`, `van der`—and ancillary information entered by photographers as a means of control—`caption correct`, `photo release`.

The percentage of accepted triplets was slightly higher than the percentage of accepted pairs. More than 72% of our triplets—i.e., 1,448 triplets—were accepted. The rest—i.e., 552 triplets—was considered unsuitable for the same reasons stated in the case of the pairs.

The main disadvantage of computing the chi-square test without removing initials, acronyms and dates is that these terms have very high chi-square values—obviously, they are more typical of Belga captions than of Europarl. Indeed, 26 of our top 50 singles were rejected for being initials, acronyms or dates. Adding acronyms and initials to our stop-word list would have ensured that they would not appear in our singles. However, the large number of press agencies and freelance photographers made it impossible for Belga staff to give us a complete list of such terms.

Figure 3 compares the efficiency of our method for the automatic extraction of keywords with that of an ideal algorithm that produces keywords that are always accepted by the annotators. The different categories of our results—singles, pairs and triplets—are represented by different curves. For the total number of keywords extracted, Figure 3 shows that our singles' efficiency curve is very close to that of the ideal algorithm.

For illustration purposes, Table 1, Table 2 and Table 3 display the 20 keywords of each category that were assigned the highest chi-square values and were accepted by the annotators. Readers interested in our results should download the full lists stored on the authors' website (`http://osiris.sunderland.ac.uk/~cs0mpl/Belga/`). The lists shown in these tables contain "stems", rather than words—for instance, the stem `minist`, displayed in Table 1, was derived from the words *minister*, *ministers*, *ministe*, *minist*, *ministeer*, *ministes* and *ministered*, which were all found in Belga captions and were grouped altogether by the stemming algorithm. Readers interested in knowing the actual words from which the stems were derived should visit the authors' website and look them up.

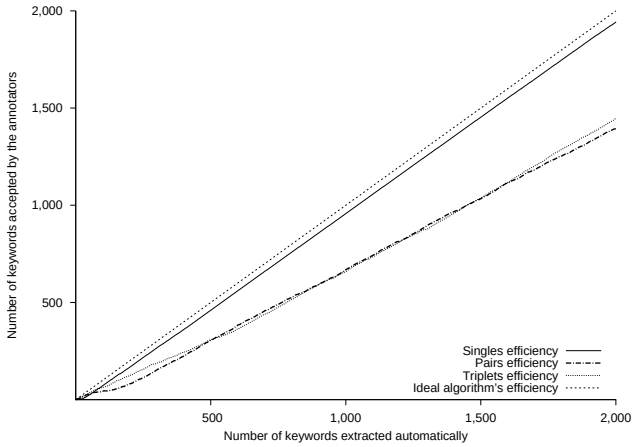


Figure 3: Efficiency curves

Note that Table 2 and Table 3 display stems too—for example, the stem `press confer` in Table 2 comes from the collocations *press conference*, *presse conference*, *press conferences*, *press conferance* and *press conferred*.

Keyword	Chi-square value
belgium	63,243.20
match	43,137.91
brussel	39,693.99
soccer	27,278.65
minist	21,035.98
team	18,273.36
cup	18,187.04
citi	17,369.72
leagu	16,693.42
celebr	16,545.52
won	16,436.22
championship	16,256.48
dai	14,198.89
germani	13,937.73
ball	13,692.43
race	13,484.38
arriv	12,429.02
tenni	12,348.60
franc	11,817.51
stadium	11,753.17

Table 1: Singles with the highest chi-square values (Belga vs. Europarl)

Our method for the extraction of keywords was very good at finding adjectives and nouns. Although we did not deliberately exclude particular parts of speech, most of our keywords contained, exclusively, adjectives and proper and common nouns. Only a few verbs appeared in our keywords and these tended to be rejected by the annotators.

8. PERFORMANCE ANALYSIS

Our code was written in Java SE 6 entirely. Experimentally, our implementation has been run on a computer with an Intel© Xeon© CPU 5150 processor with 2GB of RAM, running under Microsoft Windows XP version 2002 SP2.

Keyword	Chi-square value
brussel belgium	7,543.31
soccer match	6,040.35
leagu soccer	4,268.11
press confer	3,910.66
world cup	3,548.26
prime minist	3,372.98
grand prix	2,290.11
round match	2,050.57
world championship	1,984.63
tenni tournament	1,984.19
west bank	1,859.44
train session	1,835.15
final match	1,788.60
soccer team	1,637.71
cycl race	1,541.24
de franc	1,536.55
club brugg	1,507.01
tour de	1,482.13
champion leagu	1,478.61

Table 2: Pairs with the highest chi-square values (Belga vs. Europarl)

Keyword	Chi-square value
leagu soccer match	2,698.38
divis leagu soccer	1,071.68
tour de franc	1,043.78
georg w bush	968.54
fifa world cup	912.81
uk and ireland	912.08
open tenni tournament	906.01
belgian first divis	792.95
belgian first leagu	781.59
presid vladimir putin	645.90
nation soccer team	638.68
pope john paul	611.18
john paul ii	594.14
russian presid vladimir	585.81
formula on driver	580.35
de franc cycl	551.64
franc cycl race	538.85
king albert ii	528.34
london unit kingdom	502.18
headquart in brussel	499.28

Table 3: Triplets with the highest chi-square values (Belga vs. Europarl)

On average, the code to extract singles from a collection of 1,727,159 Belga captions requires less than 23 minutes. The average was computed after 50 executions, and the code performs the pre-processing and lexical analysis of both corpora—Belga captions and Europarl—the chi-square computation, the selection of keywords and their sorting by chi-square value.

While the execution takes place, a number of files containing statistical information are printed out. Such files display the full set of captions after having being stemmed, the whole set of stems present in the captions sorted by frequency, the actual words associated with each stem, and the final set of keywords sorted by chi-square value.

Computation times for pairs and triplets are longer than for singles: on average, the code to extract pairs requires 2 hours and 31 minutes and triplets extraction requires 6 hours and 10 minutes. The vast majority of the time spent in pairs' and triplets' extraction is consumed determining collocations and counting their frequencies of appearance.

9. BELGA QUERY LOGS

We decided to extend the automatic extraction of keywords to consider query log data, after realising the growing number of studies that are using such data to investigate aspects related to information retrieval—for instance, the 101 semantic concepts defined by Snoek *et al.* for the *MediaMill Challenge* were chosen after an extensive analysis of query logs [26]. While we do not use the logs as our primary source of information, we do compare the set of keywords derived from the corpus with those derived from the logs, and we combine them to create a comprehensive list of indexing terms that incorporates both users' and content providers' perspectives.

Belga query logs comprise a great deal of useful information about captions and pictures that could not possibly be recreated in a laboratory setting: the complete user sessions—query history and viewed, clicked and retrieved pictures—for several users in a variety of contexts.

We mined the logs for the same period when the captions that we analysed previously were published (22 June 2007 – 2 October 2007). We took into account all the queries made by the users, regardless of the results for those queries—certain queries produced no images in return, or the images returned did not satisfy the users' interest. However, we took them all into account, because we wanted to learn the full list of keywords that might be of interest for the users.

After tokenising the logs, removing stop-words and stemming the text, we recomputed the chi-square test, comparing Europarl with the queries placed by the users. In total, we extracted 10,811 singles, 10,737 pairs and 4,428 triplets. All these keywords were assigned chi-square values greater than 31.2137. For illustration purposes, Table 4 shows the 20 keywords with the highest chi-square values that were extracted from the logs. As explained above, pairs and triplets were extracted too, but their chi-square values were not as high as those of the singles. The complete list of keywords extracted from the query logs, together with the actual words from which these keywords were stemmed, are available on the authors' website [16].

It is worth noting that, in order to maintain a current index, our method for the extraction of keywords has to be recomputed regularly. Belga publishes more than 300 pictures everyday. Therefore, its set of captions and query logs are constantly evolving.

9.1 Captions vs. Logs

Keywords extracted from the query logs are expected to make useful indexing terms, because actual users have already employed them to search for content. Indeed, Belga may wish to take such information into consideration to determine the set of pictures that they publish in the future.

We were especially interested in determining the set of keywords that had significantly greater relative frequency in the logs than in the captions. It would be useful for Belga staff to know such a set, as it could help them to improve their indices to reflect users' interest.

Keyword	Chi-square value
belgium	126,062.13
leterm	86,946.85
clijster	84,767.40
soccer	72,570.86
tour	64,767.79
anderlecht	58,617.17
albert	53,854.64
rompui	41,387.42
king	39,986.91
genk	32,950.54
gevaert	32,759.11
henin	31,667.90
beckam	31,096.58
gent	30,234.70
fahi	28,115.56
cycl	27,870.08
china	25,367.50
lalmand	25,306.58
princ	23,601.41
sarkozi	22,594.95

Table 4: Keywords derived from Belga query logs vs. Europarl

After establishing the list of singles that had significantly greater relative frequency in the logs than in the captions, we computed their chi-square values, comparing logs with captions. A total of 901 of these singles were assigned chi-square values above 31.2137. For illustration purposes, Table 5 shows the first 20 of these singles.

Keyword	Chi-square value
leterm	252,579.45
rompui	204,849.66
beckam	189,611.61
clijster	103,227.33
gevaert	58,192.64
fahi	57,479.39
angelica	51,670.03
sterchel	41,886.53
milquet	35,075.16
rugbi	34,331.77
dehaen	33,960.55
diana	33,809.85
rasmussen	29,528.60
hertoginned	27,622.54
albert	27,543.79
sarkozi	22,923.27
pavarotti	21,068.19
mccann	19,382.21
damm	18,233.18
boonen	18,008.33

Table 5: Logs vs. captions: Singles

We repeated the same analysis with pairs and triplets too. A total of 73 pairs and 34 triplets with greater relative frequency in the logs than in the captions were assigned chi-square values above 31.2137. Readers are welcome to visit the authors' website for further details on this matter [16].

10. CONCLUSIONS

We have presented a method to automatically acquire a set of keywords that characterise the content of a large multimedia collection. Our method employs the chi-square test to compare captions associated with pictures with a model of general English language. The words that deviate from the model are very specific of the collection and thus make an appropriate set of keywords to represent it.

Our work has been carried out as part of the VITALAS project using data supplied by the Belga News Agency. To confirm that the keywords that we extracted were suitable to index Belga's collection of pictures, professional annotators reviewed our results manually. More than 97% of our top 2,000 one-word keywords were accepted by the annotators.

The chi-square test was also used to derive a set of indexing terms from a comparison of Belga query logs with Europarl's model of English language and Belga captions. Having originated from actual users, this set comprises keywords that users are really interested in when searching for images. We propose that a complete set of indexing terms for a professional multimedia archive can be made of the union of the approved keywords derived from picture captions and the keywords extracted from the query logs. Such a comprehensive list of indexing terms would incorporate both users' and content providers' perspectives.

In the next few paragraphs, we outline some open issues and directions for future work.

10.1 Future Work

Collocations can be either *rigid* or *flexible* [25]. Rigid collocations are those made of terms that always occur adjacent to each other and appear in the same order—for example, *the New York Stock Exchange*—whereas flexible collocations can have intervening words placed between them, or can occur in a different order, and may even allow some inflected forms—for example, a collocation linking *to make* and *decision* may appear as *to make a decision*, *decisions to be made*, *made an important decision*.

Our study was restricted to rigid collocations. However, we anticipate carrying out further work on this subject, and we plan to follow Seretan and Wehrli's method to deal with flexible collocations [23], because it does not require extensive computation or huge training collections.

Future work should also be undertaken in the area of *named entity recognition*. Thus far, we have implemented a software component that seeks for proper nouns comprised in BELGA captions assuming that they are spelled with an initial capital in all their occurrences—as opposed to common nouns, which are spelled with a lower-case initial at least once. Originally, this heuristic was suggested by Hofland and Johansson [5], and it should be improved as our work progresses.

A first execution of our named entity recognition component delivered a total of 51,028 names out of our set of 1,727,159 Belga captions. Regrettably, at the time of writing, professional annotators were not yet available to evaluate our list of names.

11. ACKNOWLEDGMENTS

This research was supported under the EU-funded VITALAS project (project number FP6-045389). The authors are very grateful to Belga News Agency for providing the data used to carry out their research.

12. REFERENCES

[1] R. K. Belew. *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*. Cambridge University Press, Cambridge, UK, February 2001.

[2] Belga. *Belga News Agency*. `http://www.belga.be/`.

[3] C. Buckley. Implementation of the SMART Information Retrieval System. Technical Report TR85-686, Computer Science Department, Cornell University, Ithaca, New York, May 1985.

[4] P. E. Greenwood and M. S. Nikulin. *A Guide to Chi-Square Testing*. Wiley-InterScience, New York, 1996.

[5] K. Hofland and S. Johansson. *Word Frequencies in British and American English*. Norwegian Computing Centre for the Humanities, Bergen, Norway, 1982.

[6] A. Hulth and B. B. Megyesi. A Study on Automatically Extracted Keywords in Text Categorization. In *Proceedings of the International Conference on Computational Linguistics*, pages 537–544, Sydney, Australia, 2006.

[7] INA. *Institut National de l'Audiovisuel*. `http://www.ina.fr/`.

[8] IRT. *Institut für Rundfunktechnik*. `http://www.irt.de/`.

[9] R. Jackendoff. *The Architecture of the Language Faculty*. The MIT Press, Cambridge, MA, December 1996.

[10] A. Juilland, D. Brodin, and C. Davidovitch. *Frequency Dictionary of French Words (The Romance Languages and Their Structures First Series F1)*. Mouton & Co., The Hague, The Netherlands, 1970.

[11] P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Machine Translation Summit*, Phuket, Thailand, September 2005.

[12] G. Leech and R. Fallon. Computer Corpora - What do They Tell Us about Culture? *International Computer Archive of Modern and Medieval English*, (16):85–100, 1992.

[13] J. L. Martínez-Fernández, A. García-Serrano, P. Martínez, and J. Villena. Automatic Keyword Extraction for News Finder. *Lecture Notes in Computer Science*, 3094:99–119, 2004.

[14] I. Mel'čuk. Collocations and lexical functions. In *Phraseology: Theory, Analysis, and Applications*, pages 23–54. Oxford: Clarendon Press, 1998.

[15] M. P. Oakes and M. Farrow. Use of the Chi-Squared Test to Examine Vocabulary Differences in English Language Corpora Representing Seven Different Countries. *Literary and Linguistic Computing*, 22(1):85–99, 2007.

[16] M. A. Palomino. *An Analysis of Belga Captions and Query Logs*. July 2008. `http://osiris.sunderland.ac.uk/~cs0mpl/Belga/`.

[17] R. L. Plackett. Karl Pearson and the Chi-Squared Test. *International Statistical Review/Revue Internationale de Statistique*, 51(1):59–72, April 1983.

[18] M. Porter. An Algorithm for Suffix Stripping. *Progam*, 14(3):130–137, July 1980.

[19] P. Rayson, G. Leech, and M. Hodges. Social Differentiation in the Use of English Vocabulary: Some Analyses of the Conversational Component of

the British National Corpus. *International Journal of Corpus Linguistics*, 2(1):133–152, 1997.

[20] J. Rupert G. Miller. *Simultaneous Statistical Inference*. Springer, New York, NY, March 1981.

[21] I. A. Sag, T. Baldwin, F. Bond, and A. Copestake. Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics* , pages 1–15, Mexico City, February 2002.

[22] V. Seretan and E. Wehrli. Multilingual Collocation Extraction: Issues and Solutions. In *Proceedings of the Workshop On Multilingual Language Resources And Interoperability*, pages 40–49, Sydney, Autralia, July 2006.

[23] V. Seretan and E. Wehrli. Collocation Translation Based on Sentence Alignment and Parsing. In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles*, pages 401–410, Toulouse, France, June 2007.

[24] F. Smadja. Xtract: An Overview. *Computers and the Humanities*, 26(5–6):399–413, December 1992.

[25] F. Smadja. Retrieving Collocations from Text: Xtract. *Computational Linguistics*, 19(1):143–177, March 1993.

[26] C. G. M. Snoek, M. Worring, J. C. van Gemert, J. M. Geusebroek, and A. W. M. Smeulders. The Challenge Problem for Automated Detection of 101 Semantic Concepts in Multimedia. In *Proceedings of ACM Multimedia*, pages 421–430, Santa Barbara, CA, October 2006.

[27] VITALAS. *Video and Image Indexing and Retrieval in the Large Scale*. `http://www.vitalas.org/`.

[28] Wikipedia. *Wikipedia:About*. `http://en.wikipedia.org/wiki/Wikipedia:About`.

Spatiotemporal Region-oriented Attentional Selection of Active Salient Objects

Thomas Geerinck, Hichem Sahli, Iris Vanhamel, Valentin Enescu
Electronics & Informatics Department (ETRO)
Vrije Universiteit Brussel (VUB)
Pleinlaan 2, 1050 Brussels, Belgium
{tgeerinc,hsahli}@etro.vub.ac.be

ABSTRACT

This paper proposes a framework, based on a spatiotemporal attentive mechanism for automatic region-of-interest determination, corresponding to events in video sequences of natural scenes of dynamic environments. We view this work as a preliminary step towards the solution of high-level semantic event analysis. More specifically, we wish to detect a visual event within a cluttered scene, without intensive training algorithms. In contrast to event detection methods used in the literature, which drive attention based on motion and spatial location hypothesis, in our approach the visual attention is region-driven as well as feature-driven. For this purpose, a two stages attention mechanism is proposed. In a first phase spatiotemporal activity analysis extracts keyframes from the image sequence and selects salient areas within these frames. The three types of visual attention features we used are intensity, color and motion conspicuity. Consequently, the selected areas are further processed to determine the most interesting active-region, based on a newly defined region saliency measure. Qualitative and quantitative results using the proposed framework are illustrated using an image sequence of a crowded train station.

Keywords

Event detection, activity measure, visual attention, active object.

1. INTRODUCTION

It is common that, we as humans shift our attention toward anything that is interesting to us. This process is a natural form of perception of event/activity detection. In general terms, an event can be defined as a qualitatively significant change in the behavior of the data, as defined by the domain user [22]. In the context of video analysis, a visual event is commonly defined based on a moving object with constraints in its size, color or shape instances that haven't been seen before [4, 19].

Our goal in this work is to formalize a strategy for efficient detection and localization of active salient region-of-interest, responsible for events in video sequences of natural scenes of dynamic environments. This issue forms the basis of what might lead in a later stage to object behavior recognition and understanding [21], which is yet an unsolved problem.

In this work, biologically inspired methods have been chosen. Attentive mechanisms are found to be relevant, e.g., for the selection of incoming visual information, for the decision making in top-down, i.e., symbolic to sensory information processing, for the selective functioning within the organization of behaviors, or for the understanding of individual and social cognition [9]. Attentive mechanism in computer vision aims at mimicking the ability of natural vision systems to select just the relevant aspects of the broad visual input, and should be considered as a set of strategies that attempts to reduce the computational cost of the search processes inherent in visual perception [20]. For modeling visual attention, it is crucial to select an appropriate set of visual descriptors, e.g. local color descriptors, color histograms, and motion descriptors, which can help establishing a connection towards the semantically meaningful features of content [16].

Conventional region-of-interest (ROI) determination based on visual attention principles, encapsulate (i) temporal and motion information which characterize the selected event, (ii) static or video-based feature combination method, and (iii) integrating saliency-oriented and task-oriented influences[8]. Motion is indeed of fundamental importance in biological vision systems and contributes to visual attention as confirmed by Watanabe et al. [17]. As such motion will be the most important cue which will be used in the proposed framework. Note that, our current implementation, considers only saliency-oriented ROI determination, meaning no top-down influences are introduced.

In our approach, events are regarded as qualitatively significant changes in the behavior of a defined motion activity measure [11, 16]. In contrast of previously defined motion activity measures, e.g. as in [16], where two simple descriptors have been used for describing monotonous activity (defined as the average block-based motion vector magnitude) and non-monotonous activity (approximation as the average temporal derivative of motion vectors), in this work we combine intensity, color, and motion conspicuity features. This allows defining motion activity as the gross, overall motion content in a given video segment, such as, high or low activ-

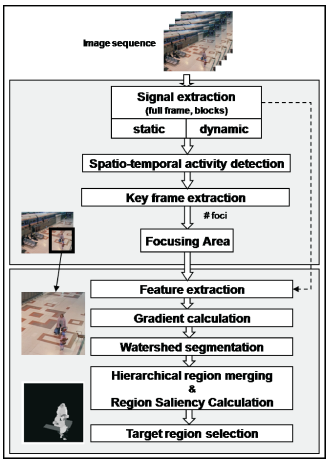


Figure 1: Global system architecture.

ity, spatially coherent or scattered activity, etc... In contrast to event/novelty detection methods used in the literature, e.g. [4] where a clustering based learning mechanism that incorporates habituation theory is used, our approach detects valuable events in cluttered and chaotic environments without intensive training algorithms.

Figure 1 depicts the proposed framework for region-based event detection. It is composed of two major modules:

- Activity detection module (Section 2.1 and 2.2), consisting of a spatiotemporal motion processing module allowing the detection of key-frames in the image sequence. The presented activity detection module augments previous work on activity detection, described in [5].
- A region-driven focus of attention module (Section 2.4). Attention allows us to focus on the relevant regions in the scene and thus reducing the amount of information needed for further processing. The extraction of active salient objects of interest, follows previous work on salient object extraction, described in [6].

Qualitative and quantitative results are given in Section 3, using an image sequence of a crowded train station. Finally, Section 4 draws some conclusions and further research development.

2. EVENT DETECTION IN VIDEO STREAMS

Consider a dynamic phenomenon, whose behavior changes enough over time so as to be considered a qualitatively significant change. Each such change is an event. An example is the change of Station Hall traffic from normal to some one running. In order to detect an event, outliers, peaks, trends or changes in trends should be spotted automatically in the available measures (data) considering a well specified time-window. Using a short time-window, instantaneous events or activities are emphasized, in general represented by outliers or sudden change in amplitude of the signal. Consequently, using a longer time-window, other time series analysis techniques can be applied to detect other types of peculiarities, such as trends, changes in trends, periodicity, etc.

In this work, we focus on detecting frame-to-frame events, possible starting points of an event, etc... Events lasting for a certain time period or during a video segment, are not considered at this stage of the development. In this section, we first define the *activity measure* used as time series data for event analysis, and second we propose methods for event detection by identifying the time points at which the behavior's of the activity measure change occurs.

2.1 Activity Measure

In the current framework only short time-windows are considered. We define a time-window T_w of approximately 3 - 4 s. Given an RGB image sequence with a frame rate f of 20 - 30 Hz, the analysis is made for $N = fT_w$ frames. Each image frame, $F_k(k = 1 \dots N)$, is divided into n quadrants, $F_k^j(j = 1 \dots n)$ (100 in the current implementation). For each quadrant j an activity measure over time, $A_k^j; k = 1 \dots N$, is estimated. Time series analysis will allow detecting the key-frame containing an event, and to localize the detected event within the frame, for further analysis.

The activity measure, within an RGB image quadrant F_k^j, includes the estimation of:

- Δh_k^j the inter-frame color histogram change,
- $\mu(\Delta \mathcal{I})_k^j$ the inter-frame mean intensity change,
- $\mu(\mathcal{M}_c)_k^j$ the mean motion conspicuity,
- $\mu(\dot{\mathcal{M}}_c)_k^j$ the mean acceleration conspicuity.

The used low-level features, intensity, color and motion conspicuity are chosen based on intuitive implications concerning perceptual behavior of humans on changing scenes:

- Possible local illumination variations might attract attention.
- The color decomposition change makes a distinction between high contrast moving foreground objects against static background and low contrast moving foreground objects against static background. The moving object with high color contrast will attract attention compared to a lower color contrasted moving object with same motion characteristics.
- Image segments with spatially consistent motion field are more likely to be parts of foreground moving objects and receive more user attention than those in the background do.
- The user's attention is drawn to objects with sudden variations in their motion pattern.
- Objects with larger motion draw more attention than those with smaller motion do.
- People can pay attention to a limited number of objects in a scene. When there are many different objects moving with similar motion characteristics, humans loose the ability to pick out the most significant ones.

The HSI color space has been chosen to provide an intuitive representation of color and to approximate the way in which humans perceive and manipulate color [12]. The used color transformation is described in detail in [5]. For the color histogram estimation, the HSI space is uniformly quantized into a total of 256 bins. This includes 16 levels for $\mathcal{H}$, 4 levels for $\mathcal{S}$, and 4 levels for $\mathcal{I}$. Finally, for an image quadrant j, the inter-frame color histogram distance, Δh_k^j , is estimated using the Euclidean distance between the color histograms at time k and $k+1$, defining a color similarity [18] between successive image frames.

The inter-frame intensity change is estimated as $|\mathcal{I}_k - \mathcal{I}_{k+1}|$; $\mathcal{I}_k$ being the intensity map.

Motion conspicuity copes with certain observations from human attentional perception concerning moving objects in dynamic scenes: (i) a slow moving objects surrounded by many fast moving objects draws attention; (ii) a left moving object surrounded by right moving objects attracts attention; (iii) a sudden motion direction variation of a moving object in a stream of consistently moving objects attracts attention; (iv) an object with increasing velocity (acceleration), amongst several objects with decreasing velocity (deceleration) draws attention.

Consider the following scenario, a video sequence is captured by a camera that is looking at a classroom entrance. At the time the class is dismissed, the majority of the students will be going out of the classroom. In this situation, if two people are trying to walk back into the room, their actions would be considered "irregular" compared to the rest of the students. Attention analysis aims at quickly highlight the abnormal regions and perform further activity analysis on these regions.

Therefore, we propose an innovative approach for identifying and selecting significant salient motion from different moving objects. The significance of motion is represented by means of a pixel motion conspicuity value $\mathcal{M}_c$.

The optical flow motion vectors $\mathcal{M}_x$ and $\mathcal{M}_y$ are estimated using the Lucas-Kanade [10] algorithm applied on the intensity map $\mathcal{I}$. From these motion vectors, motion maps are derived: concerning motion amplitude, we make a distinction of three levels (low velocity, medium velocity, high velocity); concerning motion direction we distinguish between four directions (up, down, left, right).

Let $u(x,y)$ and $v(x,y)$ be two components of the optical flow vectors $\mathbf{v}$ at point (x,y). Optical flow vector $(\mathbf{v})$ has amplitude m. Then, the motion feature maps: $\mathcal{M}_{left}$, $\mathcal{M}_{right}$, $\mathcal{M}_{up}$, $\mathcal{M}_{down}$, are obtained by:

$$\begin{array}{rcl} \mathcal{M}_{left} & = & (u < -T_t)?|u| : 0 \\ \mathcal{M}_{right} & = & (u > T_t)?|u| : 0 \\ \mathcal{M}_{up} & = & (v > T_t)?|v| : 0 \\ \mathcal{M}_{down} & = & (v < -T_t)?|v| : 0 \end{array} \quad (1)$$

where T_t is a threshold introduced for disregarding too small values. It is set to a value of 0.5.

The distinction between low, middle and high velocity is obtained by intelligently thresholding the motion information. Two thresholds T_{p1} and T_{p2} are estimated and initialized by respectively the 75% and 95% percentile of all motion values of the first frame. For each subsequently considered frame, these threshold values T_{p1} and T_{p2} are updated by combining the new motion information (new values for T_{p1} and T_{p2}) and the previous threshold values. Recursively this is noted as:

$$T_{p,k} = T_{p,k-1} + \frac{T_{p,k} - T_{p,k-1}}{k} \quad (2)$$

for $p \in \{p1, p2\}$ and k = 1 ... N.

The motion feature maps $\mathcal{M}_{low}$, $\mathcal{M}_{medium}$, $\mathcal{M}_{high}$ are then obtained by:

$$\begin{array}{rcl} \mathcal{M}_{low} & = & (m > T_t \wedge m < T_{p1})?m : 0 \\ \mathcal{M}_{medium} & = & (m \geq T_{p1} \wedge m < T_{p2})?m : 0 \\ \mathcal{M}_{high} & = & (m \geq T_{p2})?m : 0 \end{array} \quad (3)$$

where $m = \sqrt{u^2 + v^2}$, the motion amplitude of the optical flow vectors.

The fusion of these motion feature maps into the motion conspicuity map $\mathcal{M}_c$ is done by weighted average. The maps are first weighted by a unique weight function, then they are summed up, and finally normalized. In order to determine the most important maps and raise their influence, a normalization operator needs to be applied. In conformity with the approach for feature fusion presented in [3], we propose:

$$\mathcal{W}(X) = X/\sqrt{m} \quad (4)$$

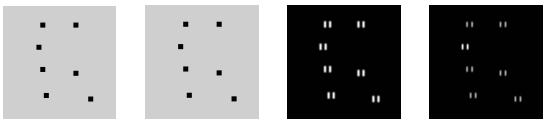
Where m is the number of local maxima in a pre-specified range from the global maximum. This ranging threshold is determined by analysis of the distribution of the maximum values, by choosing for example the median of the maxima.

The motion conspicuity map $\mathcal{M}_c$ is then obtained by combining the weighted motion feature maps $\mathcal{M}_{low}$, $\mathcal{M}_{medium}$, $\mathcal{M}_{high}$, $\mathcal{M}_{up}$, $\mathcal{M}_{down}$, $\mathcal{M}_{left}$, $\mathcal{M}_{right}$:

$$\mathcal{M}_c = \mathcal{N}(\sum_i \mathcal{W}(\mathcal{M}_i)) + \mathcal{N}(\sum_j \mathcal{W}(\mathcal{M}_j)) \quad (5)$$

with $i \in \{up, down, left, right\}$ and $j \in \{low, medium, high\}$, and $\mathcal{N}$ the normalization operator.

The effect of the uniqueness weight $\mathcal{W}$ is illustrated in Figure 2. Consider an image sequence where 5 items are moving right, and 1 item is moving leftwards. As a result, the feature map $\mathcal{M}_{left}$ is weighted higher than the feature map $\mathcal{M}_{right}$. This enables pop-out of salient motion.



(a) (b) (c) (d)

Figure 2: (a), (b) Two subsequent image frames. One object is moving left, 5 are moving rightwards, all with equal velocity. (c) The motion amplitude, equal for each object. (d) The motion conspicuity map $\mathcal{M}$, favoring the salient motion.

As in the case of motion conspicuity, the significance of acceleration is represented by means of an acceleration conspicuity value $\dot{\mathcal{M}}_c$. We distinguish between acceleration $\dot{\mathcal{M}}^+$ and deceleration $\dot{\mathcal{M}}^-$. $\dot{\mathcal{M}}_k$, is estimated as the difference between successive motion maps $\mathcal{M}_k$ and $\mathcal{M}_{k+1}$. Then, the motion difference feature maps: $\dot{\mathcal{M}}^+$, $\dot{\mathcal{M}}^-$ are obtained by:

$$\begin{array}{rcl} \dot{\mathcal{M}}^+ & = & (\dot{\mathcal{M}} > T_a)?|\dot{\mathcal{M}}| : 0 \\ \dot{\mathcal{M}}^- & = & (\dot{\mathcal{M}} < -T_a)?|\dot{\mathcal{M}}| : 0 \end{array} \quad (6)$$

where T_a is a threshold introduced for disregarding too small values. It is set to a value of 0.1.

Then, the motion difference conspicuity map $\dot{\mathcal{M}}_c$ is given by:

$$\dot{\mathcal{M}}_c = \mathcal{W}(\dot{\mathcal{M}}^+) + \mathcal{W}(\dot{\mathcal{M}}^-) \quad (7)$$

As such, for each quadrant j, the activity measure at time k, is given by:

$$A_k^j = \mu(\mathcal{M}_c)_k^j[1 + \mu(\Delta\mathcal{I})_k^j + \mu(\dot{\mathcal{M}}_c)_k^j] + \Delta h_k^j \quad (8)$$

As illustrated in Section 3, the behavior of the time series data, given by the proposed activity measure, fits our objective for detecting frame-to-frame events or possible starting points of an event. These events are represented by a peak (maxima) in the activity, or a sudden activity change. In other words, when the motion in a particular image quadrant stays monotonous over a certain time window, the measured activity will be low.

2.2 Event Analysis

The specific problem we address here is the identification of the time points at which changes occurs in the time series A_k^j. These change-points are referred to as key-frames. Note that, we consider not only the detection of key-frames, but also the most active area (focusing area) within the key-frames. The Change-Point Detection method is applied, described in [5]. A frame activity measure (by combining the quadrant's activity measure) is formed from which change-points are estimated by finding the best set of points that minimizes the error in fitting a pre-defined function. The appropriate set of points is found based on maximum likelihood method [7]. From the detected key-frames, the quadrant which has the highest activity value is selected.

2.3 Change-Point Detection

We follow the work of Guralnik and Svrivastava [7], where statistical change-point detection for time series analysis has been proposed. In standard statistical approaches, change-point detection has been made by (a) *apriori* determining the number of change-points to be discovered, and (b) deciding for the model to be used for fitting the subsequence between successive change-points. In [7] a method has been proposed for the detection of the appropriate set of number of points that minimizes the error in fitting a pre-decided function using maximum likelihood. Moreover, in [7], no constraints are imposed on the class of functions that will be fitted to the subsequences between successive change-points.

Two approaches have been proposed, the batch (offline) and the incremental (online). In the batch version the entire data set (over the whole time window) is available, from which the best set of change-points is determined. In the incremental version, the algorithm receives new data points one at a time, and determines if the new observation causes a new change-point to be discovered.

Following the notation in [7], let $A_k(k = 1, ..., N)$ be the time series to be segmented. Here k is the time variable, and A_k corresponds to the frame activity estimated as a weighted sum of the quadrants activities A_k^j $(j = 1, ..., n)$. The weight assigned to each quadrant is the standard deviation of the quadrant's activity. In the offline case, the standard deviation is measured over the considered time window. In the online case, the time window incrementally extends with the latest data point.

$$\begin{array}{rcl} \mu(A_k^j) & = & \frac{1}{N}\sum_{k=1}^{N} A_k^j \\ w^j & = & \sqrt{\frac{1}{N}\sum_{k=1}^{N}(A_k^j - \mu(A_k^j))^2} \\ A_k & = & \sum_{j=1}^{n} w^j A_k^j \end{array} \quad (9)$$

The change-points detection, is then formulated as finding a piecewise segmented model, given by

$$\begin{array}{rcl} A & = & f_1(k, \mathbf{w_1}) + e_1(k), (1 < k \leq \theta_1), \\ & = & f_2(k, \mathbf{w_2}) + e_2(k), (\theta_1 < k \leq \theta_2), \\ & = & ..... \\ & = & f_l(k, \mathbf{w_l}) + e_l(t), (\theta_{l-1} < k \leq N). \end{array}$$

Where $f_i(k, \mathbf{w_i})$ is the function (with its vector of parameters $\mathbf{w_i}$) that is fit in segment i. The θ_i's are the change-points between successive segments, and $e_i(t)$'s are error terms. Several type of basis functions can be considered, e.g. algebraic polynomials, wavelet, Fourier, etc... In our implementation, algebraic polynomials of the form given by Eq. 10 are considered.

$$p(x) = p_1 x^m + p_2 x^{m-1} + \cdots + p_m x + p_{m+1} \quad (10)$$

The reader is referred to [7] for the detailed maximum likelihood estimation method for the detection of the change-points θ_i. In our case, every change-point corresponds to a key-frame. Within each key-frame, the area with highest activity value is selected as first focusing area. The parameters vector $\mathbf{w_i}$ consist of :

- m: the polynomial degree
- p: the minimal number of points, in each segment, required for the model fitting
- δ: a user-defined threshold defining a stopping criteria in the case of the batch version, and a likelihood increase threshold for the incremental version, respectively.

As stated, every change-point corresponds to a frame of interest or key-frame. Within each key-frame, the quadrants with highest activity value are selected for further analysis. According to the local maxima in the quadrant's activity measure, certain quadrants (maximum three) are selected

together with their respective surrounding quadrants forming a rectangular region of interest of fixed size, ready for further analysis. If there are more than three key-objects in a frame, the viewer may lose his focus.

2.4 Active Object of Interest Determination

Having detected the key-image frames and corresponding quadrants with significant salient activity, the objective is to partition the selected quadrants into disjoint regions, in a manner consistent with human perception of the content, as such determining the active object of interest. For this purpose, we follow the model combining visual attention and perceptual grouping, described in [6], however, in the current case, incorporating motion conspicuity estimation into the attentional selection procedure.

2.4.1 Modeling Perceptual Grouping and Visual Attention to Active Salient Objects

On the reduced key-frame image, i.e. the selected quadrants, different feature dimensions are computed: intensity, color, orientation, and motion. For each feature, a conspicuity map is formed, which strengthens important aspects of the image and ignores others. On these conspicuity maps, a hybrid approach combining low-level saliency and region information is used to produce enhanced conspicuity maps (enhanced C maps). The C maps are enhanced with region information (from rapid and rough image segmentation), by averaging the conspicuity values in each region. As such, a multi-spectral image is obtained combining all the enhanced conspicuity maps. The different channels of the multi-spectral image are: intensity I_{enh}, orientation O_{enh}, color C_{enh}, color opponency CO_{enh}, and motion conspicuity $\mathcal{M}_{c,enh}$.

The multi-spectral image is then segmented using the watershed transformation [14]. To apply the watershed, the gradient of the multi-spectral image is obtained by combining the gradients of texture (from orientation filter responses) and the gradients of the enhanced conspicuity maps. This approach allows obtaining a final gradient capturing all perceptual edges in the input image. The processing is adapted to local properties of the image, therefore, allowing suppression of the feature gradients in textures areas, but leaving it unmodified in smooth regions [15].

The segmented regions are contributing, in a competitive manner, to a region saliency map $\mathcal{S}$ in accordance with equation (11) representing the saliency of a region R_i at the current hierarchical level h. A region's saliency is determined by its position factor, the sum of its contrast compared with the neighboring regions, its shape, its orientation conspicuity, and its motion conspicuity.

$$\mathcal{S}(R_i) = \frac{\mathcal{S}_r(R_i)}{CBI(R_i)} = \frac{CSR(R_i)OC(R_i)SI(R_i)MIr(R_i)}{CBI(R_i)} \tag{11}$$

where,

Contrast $CSR(R_i)$ is the *normalized mean color contrast* of a region with the surrounding regions, defined as $CSR(R_i) = \sum_{j=0}^{N-1} \alpha_{i,j}(\sqrt{\sum_{l\in\{L,a,b\}}(\Delta\mu_l(R_i,R_j))^2} - T_d)$, N the number of adjacent regions of R_i. $\alpha_{i,j}$ is the ratio of the length of the common boundary of R_i and R_j, over the perimeter of R_i ($\alpha_{i,j} = \frac{Length(\delta R_i \cap \delta R_j)}{Perimeter(R_i)}$). The normalization factor T_d is estimated as $T_d = \mu_d - \sigma_v$, with μ_d the mean of the color differences D_i's, and $\sigma_v = \sqrt{1/n\sum_{i=1}^{n}(D_i - \mu_d)^2}$ the standard deviation of the $n = \frac{k(k-1)}{2}$ color differences between the k generated regions after the watershed segmentation [1]. Indeed, regions, which have a high contrast with their surroundings, are likely to be of greater visual importance and attract more attention. For instance, bright colors set against a more subdued background are likely to draw the eye.

Orientation Conspicuity $OC(R_i)$ is the *orientation conspicuity* defined as the mean output value of the steerable filter (4 orientations, 3 scales) over the pixels in the region R_i, $Area(R_i)$; $OC(R_i) = \frac{\sum_{p\in R_i}\hat{O}_p}{Area(R_i)}$; $\hat{O}_p$ being the normalized orientation map (at pixel p). Indeed, orientation map is an important recognition cue, here, it is also employed to describe region orientation information importance, and calculated as defined in [3].

Shape Indicator $SI(R_i)$ is a *shape indicator* expressing the compactness of the region, defined as $SI(R_i) = \frac{perimeter(R_i)}{Area(R_i)}$. With this parameter, we try to find a trade-off between articulated regions and more compact regions of different sizes.

Prominent Motion Indicator $MIr(R_i)$ is the prominent motion indicator of region R_i. It is defined as the conspicuous motion $\mathcal{M}_c = \sum_i \mathcal{W}(\mathcal{M}_i) + \sum_j \mathcal{W}(\mathcal{M}_j)$ with $i \in \{up, down, left, right\}$ and $j \in \{low, medium, high\}$. With this parameter, we emphasize on moving objects, compared to static objects. Also, we distinct significant salient motion in the scene.

Compositional Balance Indicator $CBI(R_i)$ is the *compositional balance indicator* [2]. Let $gc(R_i)$ be the center of gravity of region R_i; $gc(R)$ the gravitational center of all regions in the image with respect to their saliency value and size, defined as $gc(R) = \frac{\sum^{regions} S_r(R_i)Area(R_i)gc(R_i)}{\sum^{regions} S_r(R_i)Area(R_i)}$; R' the region whose gravitational center is the nearest neighbor of the symmetrical point of $gc(R_i)$ with respect to the midline of the image, this as a measure of overall content of the image. Then, if $gc(R) \in R_i$, $CBI(R_i) = ||gc(R_i) - gc(R)||$, otherwise $CBI(R_i) = ||CSR(R_i)|| + ||CSR(R')|| + ||gc(R_i) - gc(R)||$. If the salient region is located near $gc(R)$, we know that the larger CSR and the nearer distance between its gravitational center and the $gc(R)$ region in the image is, the smaller CBI of the region is, meaning the higher the possibility that it will be a salient portion of the image frame. For the second case, the higher CBI (high $CSR(R_i)$ and high $CSR(R')$) shows that the image frame may balance two or more elements encouraging our eye moving between these regions. If $CSR(R_i)$ is high and $CSR(R')$ is low, than CBI will be lower, resulting in a higher saliency compared to the previous described situation, where both $CSR(R_i)$ and $CSR(R')$ are high.

As such, saliency is guided by the overall content of the image, represented inherently by the CBI factor.

The region with the highest saliency value $\mathcal{S}(R_i)$ is selected as attention region (AR). Subsequently, the segmented image is analyzed by a number of perceptual attributes based on the *mise-en-scene* principles [2].

Starting from the salient attention region (AR), a saliency driven perceptual grouping process of segmented regions is proposed to obtain perceptually meaningful regions that consist of uniform components. The goal here, is to create a hierarchy among the gradient watersheds, hereby preserving the topology of the initial watershed lines and extracts homogeneous objects of a larger scale. We apply the *waterfall* algorithm [13] for producing the nested hierarchy of partitions.

In our implementation of the waterfall algorithm, the saliency measure of a boundary is based on a collection of energy functions used to characterize desired single-region properties and pair-wise region properties. The single properties include region area, region convexity, region compactness and color variances within the region. The pair-wise properties include color mean differences between two regions and edge strength. The saliency of the boundary between two neighboring regions R_i and R_j:

$$E(\tilde{R} = R_i \bigcup R_j | R_i, R_j) = E(\tilde{R}) + E(R_i, R_j) \qquad (12)$$

Where $E(\tilde{R} = R_i \bigcup R_j | R_i, R_j)$ is the cost of merging the regions R_i and R_j, $E(\tilde{R})$ is the merged region property (saliency) and $E(R_i, R_j)$ the pair-wise property, respectively defined as follows.

$$\begin{aligned} E(\tilde{R}) &= E_{area}(\tilde{R}) \frac{1}{E_{hom}(\tilde{R})} \sum_c E_{var_c}(\tilde{R}) \\ &\quad (1 + |E_{conv}(\tilde{R})|)^{sign(E_{conv}(\tilde{R}))} \\ &\quad (1 + |E_{comp}(\tilde{R})|)^{sign(E_{comp}(\tilde{R}))} \\ E(R_i, R_j) &= E_{edge}(R_i, R_j) E_{CMDif}(R_i, R_j) \qquad (13) \end{aligned}$$

A detailed description of these energy functions is presented in [6].

Using these energy functions as region merging criteria, the saliency driven perceptual grouping process results in the formation of Active Object of Interest (AOOI), following the procedure described in [6].

3. RESULTS

The proposed framework for automatic active-region determination has been tested in an image sequence of the Munich train station provided by the Institute of microtechnology, University of *Neuchâtel*, Switzerland. A time-window (N) of 100 frames has been considered.

In order to assess our results, "ground truth" events have been defined by a specialized surveillance company, highlighted with ellipses in Figure 3. The following events have been defined. For the train station sequence (Figure 3), the person dressed in red, running in the middle of the hall in frame 50; the two tall persons appearing in the scene in the left bottom corner in frame 60 and 80; the person walking in the middle of the hall in frame 90; the two persons reappearing from behind a pillar in the middle of the scene in frame 30; the person dressed in black, reappearing from behind the publicity panel near the stairs in frame 90.

Figure 3: The test sequence of Munich train station.

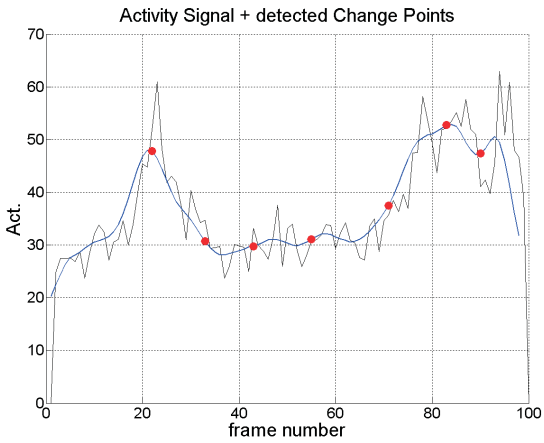


Figure 4: Change Point Detection - Batch version: Activity measure, smoothed Activity measure and detected change points.

In the following we illustrate the framework using the train station sequence, using both batch and incremental event detection strategies. Active objects of interest are only determined using the online (incremental) change point detection mechanism.

3.1 Change-Point Detection

For the Change-Point Detection experiments, a polynomial (Eq. 10) of degree $m = 4$ has been used, and the number of points needed for the model fitting in each segment has been set to $p = 5$. To facilitate the polynomial fitting procedure, the activity signal, as formulated in Equation 9, is smoothed with $\sigma = 1.5$ These values have been chosen empirically. Figure 4 and Figure 5 show the obtained results using the batch version.

Figure 6, and Figure 7 depict the results obtained using the incremental version.

Comparing the results of Figure 7 and the ground truth of Figure 3 one can notice that the proposed activity measure and the proposed change-point detection algorithm allow

Figure 5: Change-Point Detection - Batch version: Key-frames and associated active-quadrant. Below each image the frame number is indicated between brackets.

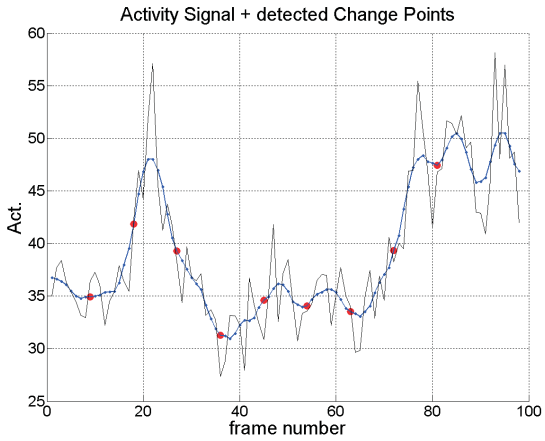


Figure 6: Change-Point Detection - Online version: Activity measure, smoothed Activity measure and detected change points.

detecting the main events in the image sequence.

3.2 Active Salient Object of Interest Determination

The proposed region-oriented visual attention based mechanism to determine and extract active objects of interest, as described in Section 2.4 has been applied to the selected quadrants in the detected key-frames, as illustrated in Figure 7. Figure 8 depicts the extracted active objects of interest.

One can notice that the segmentation process produces meaningful regions, following the shape of the perceptual object. Moreover, comparing the detected image frames (quadrants) and the extracted active salient regions to the visually selected events (Figure 3), one can notice that the proposed activity detection module with online event detection method selected (a) the running person with red jacket, (b) the two persons appearing in the scene at the left bottom, (c) the

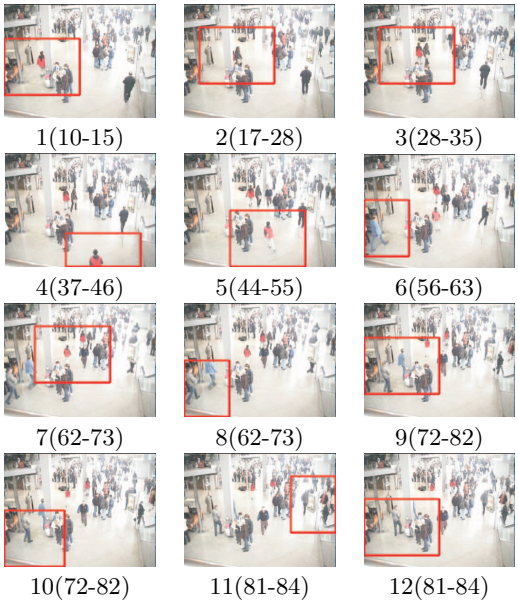


Figure 7: Change-Point Detection - Online version: Key-frames and associated active-quadrant. Below each image the winning frame number is indicated between brackets, followed by the time point of detection of the change point.

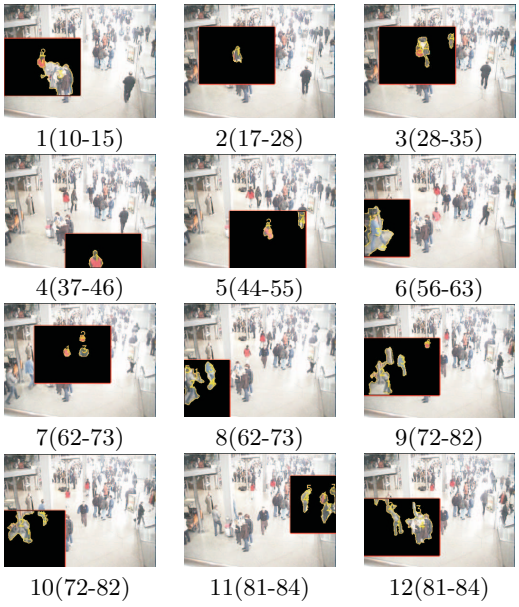


Figure 8: The extracted active objects of interest in the detected key-frames.

person near the elevator as well as (d) the two persons reappearing from behind a pillar in the middle of the scene.

4. CONCLUSIONS

In this paper we addressed the problem of activity detection in cluttered scenes. We proposed a new region-based spatiotemporal attentive mechanism. The main contributions of this work include: (i) the definition of a spatiotemporal activity measure; (ii) the use of general approaches to change-point detection, i.e. event detection that do not require training; (iii) the development of a region-based focus of attention mechanism integrating spatiotemporal features.

Comparing the results of the proposed framework with visual event (change-point) detection by humans gives promising results.

5. ACKNOWLEDGMENTS

The authors would like to thank Dr. N. Ouerhani, from the Institute of microtechnology, University of *Neuchâtel*-Switzerland, for the provided video sequence and the valuable discussion.

6. REFERENCES

[1] H. Cheng and Y. Sun. A hierarchical approach to color image segmentation using homogeneity. *IEEE Trans. on Image Processing*, 9(12):2071–2082, 2000.

[2] L. Congyan, X. De, and Y. Xu. Perception-oriented prominent region detection in video sequences. *Informatica*, 29:253–260, 2005.

[3] S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. Lecture notes in artificial intelligence (lnai), vol. 3899 / 2006, springer berlin/heidelberg, University of Bonn, 2005.

[4] R. S. Gaborski, V. S. Vaingankar, V. S. Chaoji, and A. M. Teredesai. A system for novelty detection in video streams with learning. Technical report, Laboratory for Applied Computing, Rochester Institute of Technology, Rochester, NY, USA, 2004.

[5] T. Geerinck and H. Sahli. Region-oriented visual attention-based activity detection. In L. Paletta and E. Rome, editors, *LNCS-LNAI Attention in Cognitive Systems*, pages 481–496. Springer-Verlag, 2007.

[6] T. Geerinck, H. Sahli, D. Henderickx, I. Vanhamel, and V. Enescu. Modeling attention and perceptual grouping to salient objects. In L. Paletta and E. Rome, editors, *LNCS-LNAI Attention in Cognitive Systems*, page tbp. Springer-Verlag, 2008.

[7] V. Guralnik and J. Srivastava. Event detection from time series data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–42, New York, NY, USA, 1999. ACM Press.

[8] Y. Hu, X. Xie, W.-Y. Ma, L.-T. Chia, and D. Rajan. Salient region detection using weighted feature maps based on the human visual attention model. In *2004 Pacific-Rim Conference on Multimedia (PCM 2004)*, Tokyo Waterfront City, Japan, November 2004.

[9] L. Itti. *Models of Bottom-Up and Top-Down Visual Attention*. PhD thesis, California Institute of Technology, 2000.

[10] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *In Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981.

[11] S. K. Makrogiannis and N. G. Bourbakis. Motion analysis with application to assistive vision technology. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)*, pages 344–352, 2004.

[12] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions On Circuits And Systems For Video Technology*, 11(6):703–715, Jun 2001.

[13] B. Marcotegui and S. Beucher. Fast implementation of waterfall based on graphs. In C. Ronse, L. Najman, and E. Decenciere, editors, *Mathematical morphology: 40 years on; proceedings of the 7th international symposium on mathematical morphology*, volume 30 of *Computational imaging and vision*, pages 177–186, Fontainebleau-France, Apr 2005. Springer.

[14] F. Meyer. An overview of morphological segmentation. *IJPRAI*, 15(7):1089–1118, 2001.

[15] R. O'Callaghan and D. Bull. Combined morphological-spectral unsupervised image segmentation. *IP*, 14(1):49–62, January 2005.

[16] K. Peker, A. Alatan, and A. Akansu. Low-level motion activity features for semantic characterization of video. In *Multimedia and Expo, 2000. ICME 2000.*, volume 2, pages 801–804, New York, NY, USA, 2000. 2000 IEEE International Conference on Publication.

[17] K. Rapantzikos and N. Tsapatsoulis. Enhancing the robustness of skin-based face detection schemes through a visual attention architecture. In *ICIP05*, pages II: 1298–1301, 2005.

[18] J. R. Smith and S.-F. Chang. Tools and techniques for color image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 426–437, 1996.

[19] A. Tentler, V. Vaingakar, R. Gaborski, and A. Teredesai. Event detection in video sequences of natural scenes. Technical report, Rochester Institute of Technology, Laboratory for Applied Computing, 2002.

[20] J. Tsotsos. Motion understanding: Task-directed attention and representations that link perception with action. *International Journal of Computer Vision*, 45(3):265–280, 2001.

[21] J. K. Tsotsos. Distributed saliency computations solve the feature binding problem. In L. Paletta, J. K. Tsotsos, E. Rome, and G. W. Humphreys, editors, *WAPCV2004: 2nd international workshop on attention and performance in computational vision*, 2004.

[22] C. Varadharajan. *A Wavelet-Based System for Event Detection in Online Real-time Sensor Data*. PhD thesis, Massachusetts Institute of Technology, 2004.

Exemelification of Parliamentary Debates

Tim Gielissen and Maarten Marx
ISLA, University of Amsterdam
Kruislaan 403 1098 SJ Amsterdam, The Netherlands
maartenmarx@uva.nl

ABSTRACT

Parliamentary debates are an interesting domain to apply state-of-the-art information retrieval technology. Parliamentary debates are highly structured transcripts of meetings of politicians in parliament. These debates are an important part of the cultural heritage of countries; they are often free of copy-right; citizens often have a legal right to inspect them; and several countries make great effort to digitize their entire historical collection and open that up to the general public. This provides many opportunities for the IR community.

In this paper we analyze the structure of the parliamentary proceedings and sketch a widely applicable DTD. We show how proceedings in PDF format can be transformed into deeply nested XML. We call this process "exemelification". Having the proceedings in XML makes a wide range of applications possible. We elaborate on four of these: entry point retrieval, advanced content and structure search; automatic creation of tables of contents and hyperlinked navigation menus; large savings on storage space and bandwidth for scanned documents.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous; D.2 [**Software**]: Software Engineering

Keywords

XML, Semi Structured Data, Democracy, Information Retrieval, Information Extraction

1. INTRODUCTION

Parliamentary proceedings are an interesting set of data to apply state-of-the-art information retrieval technology. Parliamentary proceedings are written records of parliamentary activities containing a wide range of document types. In this paper we only discuss notes of meetings of parliament. As with all meeting notes, these records have the purpose to store the content of the meeting. They have varying degrees of detail. Currently in most Western democracies it is common to transcribe everything that is being said, keeping the content, but making it grammatically correct and pleasant to read.

We list a number of characteristics which make these documents of special interest to the IR community:

- large historical corpora; For example, in Holland all data from 1814 will be available in 2010, at the time of writing it is available since 1974; for the Flemish parliament all data since 1971 is available in PDF; the British Hansard archives have all parliamentary minutes since 1803 available in XML.
- documents contain a lot of consistently applied structure which is rather easy to extract and make explicit;
- transcripts of meetings might be accompanied by audio and video recordings, creating interconnected multimedia data [13];
- data integration issues and opportunities [8, 4, 9] both within one country (collections from different periods, in different formats, styles, language, ...), and across countries (Cross-lingual IR);
- natural corpus for content and structure queries, combining keyword search with XPath navigation and selection [6, 11];
- natural corpus for search tasks in which the answers do not consist of documents: *expert* or *people search* [1], video search[1] and *entry point retrieval* [14].

From this list, this paper treats the information extraction, data integration and entry-point retrieval aspects. The paper is organized as follows: Section 2 describes the structure of parliamentary meetings and formalizes it in a DTD. Section 3 describes the techniques used in the exemelification process. We discuss four benefits of exemelified data in Section 4 and conclude in Section 5.

A search engine containing all Dutch parliamentary data from 1984 till May 2008 is built and can be used at `http://www.polidocs.nl`. The corpus of over 80.000 XML files is avaliable for research on request.

[1]As done in the TRECVID workshop: `http://www-nlpir.nist.gov/projects/trecvid/`

DIR2009 Enschede

2. STRUCTURE OF PARLIAMENTARY PROCEEDINGS

Notes of a formal meeting with an agenda (e.g., business meeting, council meeting, meeting of the members of a club, etc) are full of implicit structure and contain many common elements. The notes of meetings with a large historical tradition, like parliamentary debates, are in a uniform format which fluctuates little in time. This makes these notes very well suited for text-mining.

Up to our knowledge there is at the time of writing no DTD or markup language for meeting notes available[2].

Transcripts of a meeting contain three main structural elements:

the topics discussed in the meeting (the agenda);

the speeches made at the meeting: every word that is being said is recorded together with 1) the name of the speaker, 2) her affiliation and 3) in which role or function the person was speaking;

non verbal content or actions These can be:

- list of present and absent members;
- description of actions like *applause by members of the Green Party*;
- description of the outcome of a vote;
- the attribution of reference numbers to actions or topics;
- and much more.

The analogy with the structural elements in theatrical drama is striking: scenes, speeches and stage-directions are the theatrical counterparts of the three elements just listed. These are prominent elements in the XML version of Shakespeare's work.[3] The close relation between politics and drama is an emerging theme in political science, see e.g., [5, 3].

These elements are structured as follows:

```
meeting          ⟶  (topic)+
topic            ⟶  (speech | stage-direction)+
speech           ⟶  (p | stage-direction)+
p                ⟶  (#PCDATA | stage-direction)*
stage-direction  ⟶  (#PCDATA).
```

All elements contain metadata stored in attributes. The British digitized debates from 1803 till 2004 are available in XML[4] and basically have this structure.[5]

[2]The DTD of the XML versions of the British Hansard is effectively just a container to store the text, and not suitable as a genuine model of meeting notes.

[3]`http://metalab.unc.edu/bosak/xml/eg/shaks200.zip` One of the referees pointed out the well-documented DTD for drama which is part of the TEI guidelines for text markup (`http://www.tei-c.org/release/doc/tei-p5-doc/en/html/DR.html`). This DTD is a good starting point for modelling, but for our purposes both too general and too specific.

[4]`http://www.hansard-archive.parliament.uk/`

[5]The structure though is flat instead of nested as it is here, which makes retrieval quite cumbersome. For instance, to retrieve all text spoken by MP `X` we must collect all following siblings of the `member` element which contains the name `X` which come before the next `member` element. We note that this is an example of an until-like query which is not expressible in Core XPath 1.0 [10].

Within the Dutch proceedings however there is an intermediate structural element —the block— which distinguishes the theater drama from the political debate. In Dutch parliament, the debate on each topic is organized as follows: each party may hold a speech by a member standing at the central lectern; other members may interrupt this speech; the chairman can always interrupt everyone. Most often, when all parties had their say at the central lectern, a member of government answers all raised concerns while speaking from the government table and again he or she can be interrupted. In most cases this concludes a topic, but variations are possible and occur (e.g., several members of government speaking or a second round of the whole process).

The *block* is an important debate-structural element because it indicates who is being attacked by the interrupters. Thus for the Dutch situation the DTD becomes

```
topic  ⟶  (block)+
block  ⟶  (speech | stage-direction)+
```

If this block structure is not present in meeting notes, then each topic will have exactly one block child. Thus both types of meeting fit this DTD.

Note. For presentation purposes, the DTD presented here is the core of the model. The DTD actually used contains additional elements and attributes for storing all kinds of metadata. Up till now, DTD is expressive enough for the structure that we want to capture. But we need the possibility of XML Schema to constrain data-types like dates.

Figure 1 contains a visualization of a one-topic debate which uses the block structure and which is created with an XSL-stylesheet from the XML. Each row stands for one block and each vertically positioned mouth stands for one speech. The size of the mouth is proportional to the length of the speech measured in number of words. The speaker on the central lectern has the red mouth, the interrupters have a blue mouth. Interruptions by the chairman are not shown.

We end this section with two more observations on interesting structure in debates, also visible in Figure 1:

1. Blocks consist either of one uninterrupted speech or they have the form `(red,blue)+,red`, that is a sequence of pairs of speeches by the central speaker and an interrupter ended by the central speaker.

2. Zooming in on a block, if `A` is the speaker at the lectern and `B,C,D` are the ones interrupting `A`, then blocks very often look like `(AB)+(AC)+(AD)+A`, i.e., a sequence of small conversations with different members with `A` having the last word.

Debates in the Dutch parliament are governed by a set of written regulations and a set of unwritten codes. Both observations above are instantiations of unwritten codes. The first of the rule that the speaker at the lectern always has the last word. The second of the rule that a member of parlement can only have one block of interruptions of a member at the central lectern. See [15] for these rules. Another rule is that someone may only interrupt another 3 times in a row. So according to these unwritten codes the second regular expression should be `(AB){1,3}(AC){1,3}(AD){1,3}A` and none of `B,C,D` should be equal.

Formalizations of these written and unwritten rules in terms of regular expressions, and using these to find *violations* is an interesting open direction of research.[6]

This internal structure of blocks can be used to create high-level overviews of debates which show who attacks who and which can be used for navigation. We present an example in Section 4.3. The regular expression which best fits or describes a block can be obtained by the algorithm which induces DTD's from a set of example XML-files described in [2].

3. EXEMELIFICATION: FROM FLAT PDF TO DEEP XML

Figure 2 gives a good indication of the mappings created in the exemelification process. The following technique is used. First we extract the text from the PDF using the open source program `pdftohtml`[7] with the `-xml` option. This yields an XML file with for each line of text four coordinates which indicate the bounding box of that text. Multiple columns are detected and preserved. Some font and layout information is preserved but not all. The XML structure is simple and flat:

```
root  ⟶  (page)*
page  ⟶  (text)*
text  ⟶  (#PCDATA,b,i)*
```

On these XML files we use patterns written as regular expressions to add special empty XML elements on places where in the final file an XML element needs to be opened. For instance, the □ is replaced by `<blockstart/>`. A phrase like

Mevrouw **Swenker** (VVD):

is replaced by

```
<speechstart speaker='Swenker' party='VVD' ... />,
```

with the ... containing additional information.

The result of this search and replace process is again a well formed XML file with a similar flat structure as before. In the last step we perform a cascade of groupings starting with the elements which need to be most deeply nested: the paragraphs `p`. XSLT 2.0 has a very useful command for this task: `xsl:for-each-group`. This command, new in XSLT 2.0, replaces the so-called Muenchian method which was needed in version 1.0 of XSLT [7].

4. APPLICATIONS OF THE XML STRUCTURE

We describe four applications of the XML structure. None of these is possible when working with the PDF data. They are entry point retrieval and the use of permalinks, complex content and structure queries, automatic creation of tables of contents and navigation menus and finally savings on bandwidth.

[6]We have found such violations with Dutch members of parliament who have a new debating style like Wilders and Verdonk.

[7]`http://pdftohtml.sourceforge.net/`

4.1 Entry point retrieval and permalinks

The most natural answer unit in a retrieval system for parliamentary debates is the speech. The result page after a keyword query then will be a ranked list of items consisting of

- the name of the speaker,
- her party,
- a photo of the speaker,
- the date of the speech
- a relevant text snippet of the speech,
- a hyperlink which points to the anchor attached to the speech within a debate, and
- a hyperlink to the original PDF source.

This is how it works in the UK on the site `http://www.theyworkforyou.com`, on the site of the European Parliament, and also in the retrieval engine that we built for the Dutch data `http://www.polidocs.nl`, see Figure 3.

Though natural, this notion of answer is by no means standard for parliamentary retrieval systems. The search systems of the German and Flemish parliaments return the proceedings of one day. These can be PDF files with two columns of up to a 100 pages. In the Netherlands, the situation is even more complex:

- proceedings before 1995 are available at `http://www.statengeneraaldigitaal.nl/`. The answer unit is the proceedings of a complete meeting;
- proceedings after 1995 are available at `http://parlando.sdu.nl/cgi/login/anonymous`. The answer unit roughly corresponds to one topic. It is indeed roughly as topics almost never start at the top of a page nor finish at the bottom of a page, and the PDF documents at Parlando are divided into overlapping sets of pages;
- preliminary proceedings are available at `http://www.tweedekamer.nl/`. Search is not really possible on this site. Preliminary proceedings are available in HTML which is shown together with a navigation menu which contains the same topic–block–speech hierarchy as described in Section 2.

During the transformation from PDF to XML we add a unique anchor ID to every speech. This anchor together with the number of the document given by the parliament constitutes a unique permanent reference to each speech.

The permanent hyperlinks (permalinks) for each speech made in parliament have many applications besides making entry point retrieval possible. Examples are easy referencing in emails, weblogs and even scientific papers. Permalinks also stimulate third party development of websites (like mashups) based on this data.

4.2 Complex content and structure queries

The explicit XML structure allows one to formulate information needs using natural XPath, XQuery, XSLT or NEXI [6, 11] expressions. We illustrate this by some examples:

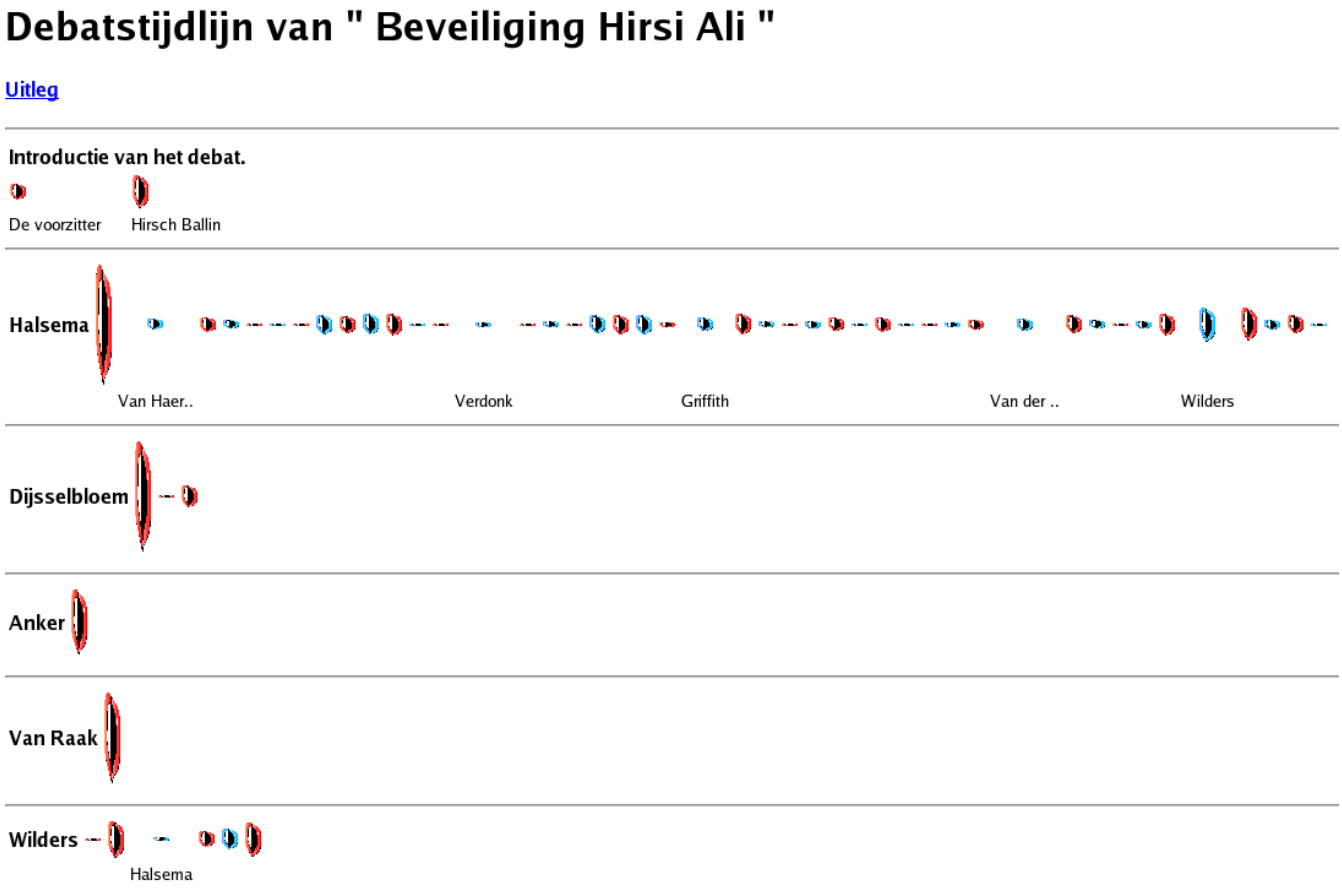


Figure 1: High-level visualization of the first part of the debate on the protection of Hirsi-Ali. Original available at `http://www.geencommentaar.nl/parlando/index.php?action=doc&filename=HAN8183A16`. The first speaker on the lectern is *Halsema* who is interrupted by *Van Haersma Buma, Verdonk, Griffith, Van der Staai* and *Wilders*, in that order. Only the first time a speaker interrupts, her name is shown.

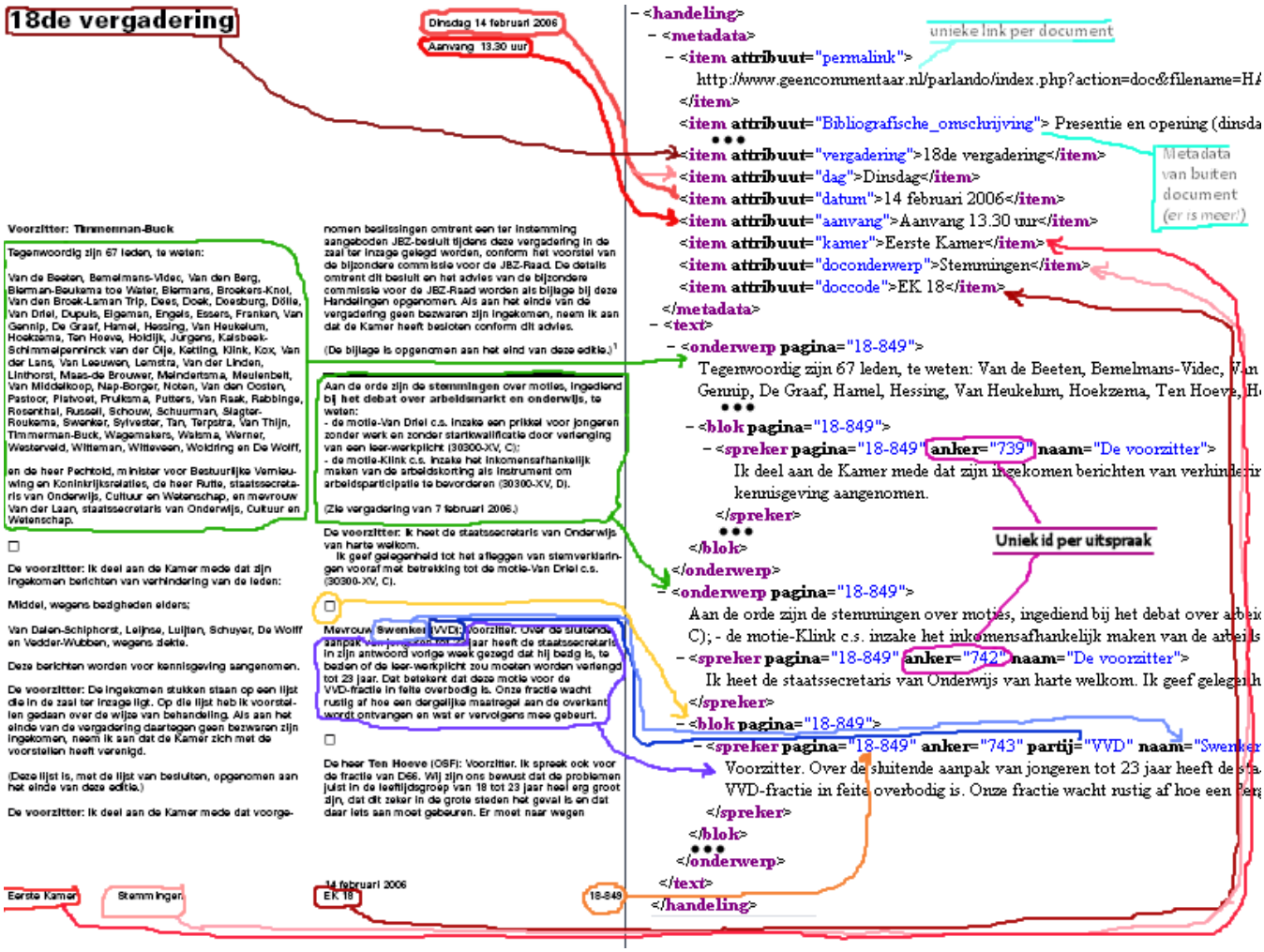


Figure 2: Example of the mapping from the description of a debate in PDF to the version in XML. Note how the start of a new block is indicated by a □ (mapping indicated in yellow).

Figure 3: Answer snippet from result list: photograph of the speaker linking to his bio, logo of his party, a link to the official PDF source, the first 100 characters of his speech and a link to the speech.

- *give speeches about Islam from debates about immigration* can be formulated as the NEXI query `//topic[about(.,immigration')]// speech[about(.,'islam')]`.
- *give all speakers who interrupted Geert Wilders during the Islam debate* can be formulated in XPath 1.0 as `//topic[@title='islam']//block[@speaker='Wilders'] //speech[@speaker != 'Wilders']/@speaker`.
- *give a list of these speakers together with their number of interruptions ordered by that number* is expressed in XQuery or XSLT using the above XPath expression and the `fn:count()` function.
- *Create a cross table of speakers at the lectern and their interrupters and list the number of interruptions in each data cel* is a typical task for XSLT. The result for the *Algemene Beschouwingen* on September 17 2008, containing 624 speeches in one debate, is reproduced in Figure 4.

Based on experience with bachelor information science students we claim that it is easier to formulate such complex queries in XSLT directly on the original XML files than to state them in SQL on a relational representation of a debate.

4.3 Automatic creation of tables of contents and navigation menus

The notes of a one day meeting of Parliament tend to be quite long, typically between 50 and 100 pages two column PDF. Within the current search engine at `www.statengeneraaldigitaal.nl` these are the documents returned to users. Unfortunately these documents do not contain a table of contents listing the topics discussed in a meeting. But even if such tables would be available in PDF they would be of little help when browsing these documents on a computer because they do not contain hyperlinks.

Since the topics are explicit elements in the XML version of the data it is straightforward to automatically generate a hyperlinked table of contents for each document. This can be done with XSLT.

Even one topic can be quite long. For instance, the meeting of September 18, 2008 took the whole day, consisted of 624 speeches with a total of 74068 words, all within one topic. Fortunately the block structure can be used to break up this large chunk of text. In fact the debate timelines in Figure 1 are navigation menus: each mouth contains a hyperlink to exactly that part of the proceedings which record the speech represented by the mouth. Again this is possible due to the added anchors.

4.4 Savings on bandwidth

The Dutch parliamentary data from before 1995 was only available in printed form. Within the StatenGeneraalDigitaal project of the Dutch Royal Library this data is scanned and OCR-ed, resulting in complex PDF documents consisting of facsimile images of every page, the OCR-ed text and a mapping from each word to its position on every page.[8]

Such files can be enormous in size. For instance, the proceedings on `http://resolver.kb.nl/resolve?urn=sgd:mpeg21:19851986:0000761` are 72 pages PDF. The size of this file is 24 Megabyte. The same proceedings in XML is less than .5Mb. We experimented with reducing the size with gzip: the PDF became 23Mb and the XML was reduced to 156Kb. This is 0.65% of the size of the original PDF.

Preliminary experiments show that using XSLT and LaTeX the original format of the proceedings can be produced with very good layout accuracy and very fast. The resulting PDF is again less than .5Mb. Producing this PDF from the gzipped XML can even be done at query time: on a standard Linux box this process took less than 1.5 seconds real time. For detailed information on this experiment see `http://ilps.science.uva.nl/PoliticalMashup/2008/10/trading-space-for-time`.

Thus large savings in bandwidth and storage space become possible. We must note that the XML version is based on OCR-ed data and contains quite a few OCR errors. Of course these come back in the PDF created from the XML source. Repairing such mistakes automatically has been done with promising accuracy by Martin Reyneart using his TICL technique [12].

We believe that the facsimiles need to be available as the ultimate source but that in a search and browse interaction process with the data the alternative, much smaller, version based on the XML is preferable. Users get results faster, they get clean hyperlinked files, and they use much less bandwidth. Once a user knows exactly which document she wants to consult, the large facsimile PDF can be downloaded.

5. CONCLUSIONS

We have shown that text extraction from Parliamentary proceedings based on regular expressions and XSLT is feasible, scalable, possible on both digital and scanned data, and leads to numerous benefits.

We stress that this extraction process is transparent, repeatable and independent of any software or hardware because we only use declarative programming languages with a well described semantics. This means that when the extraction scripts (which are themselves XML files, since it is XSLT) together with a copy of the XSLT reference [7] are stored together with the original digitized data in a safe place, it is in principle always possible to recreate the XML versions we have described here.

Several parliaments are digitizing their complete historical data. We are aware of efforts in the UK, Ireland, Australia, and the Flemish Parliament. Our DTD is general enough to fit all these proceedings. This opens the possibility of creating a huge integrated multi-lingual XML repository of parliamentary proceedings. Such a repository will facilitate comparative parliamentary (historical) research.

Acknowledgments

This research was supported by the Netherlands Organization for Scientific Research (NWO) under project number 380-52-003.

Many people have helped with creating the `www.polidocs.nl` infrastructure, in particular Robert Jan de Groot, Marina Lacroix, Breyten Ernsting and Marc Makkes. We thank the people from the Dienst Informatievoorziening from the Tweede Kamer der Staten Generaal, in particular Ben van der Beek; those from the StatenGeneraal project at the

[8] See `http://www.statengeneraaldigitaal.nl/backgrounds.html` for extensive information on the digitization process (in Dutch).

Op de spreekstoel	Achter de interruptiemicrofoon											Voorzitter	Totaal
	Kant	Van Geel	Rutte	Hamer	Wilders	Slob	Halsema	Pechtold	Thieme	Van der Vlies	Verdonk		
Kant	14	5	-	10	3	-	1	3	-	-	-	3	39
Van Geel	18	28	10	-	4	-	8	10	8	4	4	14	108
Rutte	-	14	35	17	-	9	13	-	4	-	-	9	101
Hamer	24	-	4	46	-	-	6	20	-	2	7	11	120
Wilders	-	5	-	3	11	2	11	6	-	-	-	7	46
Slob	-	-	-	-	-	5	-	10	-	-	4	6	25
Halsema	-	-	-	-	-	2	1	2	-	-	-	3	8
Pechtold	-	-	-	4	-	5	3	8	-	-	-	7	27
Thieme	-	-	-	-	-	-	1	-	-	-	-	-	1
Van der Vlies	-	-	-	-	-	-	-	1	3	2	-	3	9
Verdonk	-	-	-	-	-	-	-	3	-	-	3	2	8
Totaal	56	52	49	80	18	23	44	63	15	8	18	65	492

Figure 4: Who attacks who in the debate *Algemene Beschouwingen* on September 17 2008. Speakers at the lectern are listed in the first column; their attackers on the top row. The numbers in the cel indicate how often the person on the x-axis interrupted the speech by the person on the y-axis. The numbers on the diagonal (in gray) are the number of answers to interruptions given by the speaker on the lectern. Source: `http://staff.science.uva.nl/~marx/politicalmashup/AB2008/DebatstructuurAB2008.html`.

Koninklijke Bibliotheek, Huibert Crijns and Tineke Koster, and Hans Nielen from PDC.

6. REFERENCES

[1] K. Balog. *People Search in the Enterprise*. PhD thesis, University of Amsterdam, June 2008.

[2] G. J. Bex, W. Gelade, F. Neven, and S. Vansummeren. Learning deterministic regular expressions for the inference of schemas from xml data. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 825–834, New York, NY, USA, 2008. ACM.

[3] M. Hajer. Setting the stage, a dramaturgy of policy deliberation. *Administration & Society*, 36(6):624–647, 2005.

[4] A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In U. Dayal, K.-Y. Whang, D. B. Lomet, G. Alonso, G. M. Lohman, M. L. Kersten, S. K. Cha, and Y.-K. Kim, editors, *VLDB*, pages 9–16. ACM, 2006.

[5] R. Hariman. *Political style. The artistry of power.* University of Chicago Press, 1995.

[6] J. Kamps, M. Marx, M. de Rijke, and B. Sigurbjörnsson. Articulating information needs in XML query languages. *ACM Trans. Inf. Syst.*, 24(4):407–436, 2006.

[7] M. Kay. *XSLT 2.0 3rd edition Programmer's Reference*. Wrox, 2004.

[8] M. Lenzerini. Data integration: A theoretical perspective. In *Proc. PODS*, pages 233–246, 2002.

[9] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogeneous information sources using source descriptions. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *VLDB*, pages 251–262. Morgan Kaufmann, 1996.

[10] M. Marx and M. de Rijke. Semantic Characterizations of Navigational XPath. *ACM SIGMOD Record*, 34(2):41–46, 2005.

[11] R. A. O'Keefe and A. Trotman. The Simplest Query Language That Could Possibly Work. In *Proceedings of the 2nd INEX Workshop*, 2004.

[12] M. Reynaert. Non-interactive OCR post-correction for giga-scale digitization projects. In *Proceedings. CICLing (Computational Linguistics and Intelligent Text Processing, 9th International Conference)*, pages 617–630, 2008.

[13] J. Seaton. The Scottish Parliament and e-democracy. *Aslib Proceedings: New Information Perspectives*, 57(4):333–337, 2005.

[14] B. Sigurbjörnsson. *Focused information access using XML element retrieval*. PhD thesis, University of Amsterdam, 2006.

[15] C. van Baalen and A. Bos. In vergadering bijeen. Rituelen, symbolen, tradties en gebruiken in de Tweede Kamer. In *Jaarboek Parlementaire Geschiedenis 2008*. Boom, 2008.

Matching Queries to Frequently Asked Questions: Search Functionality for the MRSA Web-Portal

Almer S. Tigelaar & Rieks op den Akker
{a.s.tigelaar, infrieks}@cs.utwente.nl
Database & Human Media Interaction Groups,
Computer Science Department,
University of Twente

Fenne Verhoeven
f.verhoeven@utwente.nl
Technical & Professional Communication Group,
Behavioural Sciences Department,
University of Twente

ABSTRACT

As part of the long-term EUREGIO MRSA-net project a system was developed which enables health care workers and the general public to quickly find answers to their questions regarding the MRSA pathogen. This paper focuses on *how* these questions can be answered using Information Retrieval (IR) and Natural Language Processing (NLP) techniques on a Frequently-Asked-Questions-style (FAQ) database.

Keywords

Information retrieval, question-answering, linguistic processing, performance evaluation.

1. INTRODUCTION

Methicillin-Resistant *Staphylococcus aureus* (MRSA) is a strain of pathogens that is resistant to common antibiotics and is therefore hard to combat. It forms a significant threat to people with a weakened immune system.

The MRSA web-portal was developed to provide information to health care workers and the general public regarding MRSA. It is actively used by several Dutch and German hospitals and also publically accessible[1]. This research has been conducted to support this portal.

The underlying databases consist of a set of reference questions that have been collected by a field investigation and answers to these questions written by expert microbiologists [22]. The web-interface enables users to browse through the questions categorically and has a search textfield in which users can enter a query. This research focuses on providing the underlying functionality for this textfield as a component named the MRSA-QA system.

The system utilises four domain-bound question-answer sets: Dutch/Professional, Dutch/Public, German/Professional, and German/Public. The two sets tailored for the professional domain each consist of about 160 pairs whereas the two sets geared towards the general public consist of approximately 220 pairs. The questions can be viewed as perfect indices to the answers resembling the set-up of a FAQ [5].

The domain of MRSA-QA is clearly demarcated and can be classified as a *Restricted Domain Question Answering* (RDQA) system [7]. The underlying database is explicitly structured. The answers consist of several fields: a title, short guideline, instruction video, long comments section, sources, examples and additional keywords. Semantic information is represented via links in the database implicitly such as categorisation, connections between questions and answers, and links between answers and other relevant answers. The system should be able to adjust to new data, since the database can change and grow over time.

The problem can be stated as: given a query in the form of a question or a set of keywords, the system has to display a list of appropriate answers ranked by relevance in descending order. Based on this problem statement the research question can be phrased as follows:

'How can the user be provided with the most appropriate answer(s) for his or her query within the restricted MRSA domain given the available structured MRSA corpus?'

This paper focuses on answering that question by looking at and evaluating a range of techniques using the contents of the MRSA corpus. While the results are corpus specific, the methods used are generic and likely to be useful for other RDQA systems.

2. DATA

Separate datasets exist for each domain (Professional and Public) and language (Dutch and German) combination. So, there are four in total. All of these are stored in databases exhibiting the same structure, shown in Figure 1.

Starting at the top: the categories table clusters related questions for example all questions that have something to do with treatment or discharge. Each question can belong to one or more categories and points to one answer in the database. Answers can be pointed to by one or more questions and each answer may point to other related answers thereby establishing semantic links.

The answers contain the most information that can be exploited for indexing and matching in the form of fields. The title field of the answer is usually a reformulation of a related question. The guideline and comment fields are the most relevant for matching, since they contain the most free-form text with content that actually answers a question. Sources and examples are less relevant fields. Both of them contain mostly links to external documents with more clari-

[1] `http://www.mrsa-net.nl/`, `http://www.mrsa-net.nl/de`

DIR 2009, 9^{th} Dutch-Belgian Information Retrieval Workshop,
February 2–3, 2009, University of Twente, Enschede, The Netherlands.

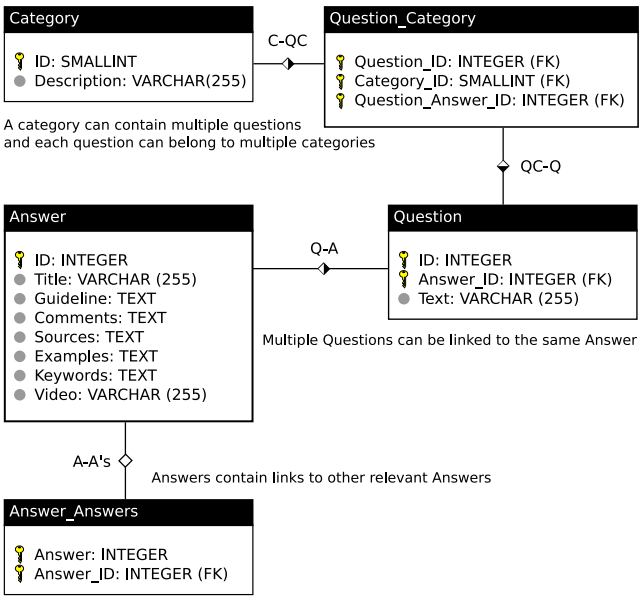


Figure 1: *Structure of an MRSA database (diamonds represent one-to-many (black/white) and many-to-many (white) relations).*

fication and are of use only when a relevant answer is being displayed. The keywords field contains a list of non-obvious keywords related to this answer and as such conveys additional semantics. The instruction video refers to non-text data. Hence, it will not be considered further.

Based on the data characteristics, we define a *document* within the MRSA-QA system as follows:

'A document consists of the title, guideline, comment and keywords of an answer plus the texts of all questions that refer to the answer.'

3. TECHNIQUES

Three categories of techniques can be distinguished: pre-processing techniques affecting the indexing and matching process, the ranking techniques and post-processing techniques which also affect the ranking process, but do not require any additional operations while indexing. The performance of six pre-processing techniques, two ranking techniques and three post-processing techniques was investigated. This section briefly explains each of the techniques.

3.1 Pre-Processing Techniques

McNamee & Mayfield and others report increases in matching performance using *character n-gramming*. In this technique a window of n characters is slid across a document. A snapshot is made of the characters that are visible in each position and used as index. There are two different approaches: n-gram each word in the text individually, called *within-word* character n-gramming, or also slide across the word boundaries in the text, called *between-word* character n-gramming. We chose for the latter approach, since it properly captures inter-word relations [8, 12].

The example presented in Table 1 provides some intuition for n-gramming. The value for n that was used is four. First the input document is padded with $n-1$ spaces on both sides (so all character sequences are properly captured). After this a window of length four is slid across the padded input text

Table 1: *Character n-gramming (spaces shown as underscores).*

Padded Input: _ _ _ _ t h e _ f o x _ _ _ _				
_ _ _ t	_ _ t h	_ t h e	t h e _	h e _ f
e _ f o	_ f o x	f o x _	o x _ _	x _ _ _

and all snapshots are recorded as shown in Table 1.

This process is applied to all documents in the corpus while indexing and on the query as well. The index thus becomes huge, since there are $k - n + 1$ n-grams for a document with k characters. Padding adds another constant $2 \cdot (n-1)$ to this. This is the primary argument against using n-grams, since such large indices adversely affect performance. For this method a value for n of five was used for both Dutch and German languages. We will show later why this is an optimal choice. Note that we use the n-grams *as replacement* of the original words they were derived from.

Word n-gramming bears much resemblance to character n-gramming. Here instead of regarding characters as the most atomic unit, words are used. While word n-gramming also leads to a larger indexing table with respect to bag-of-words indexing, the size increment is clearly not as big as with character n-grams. A value of n of two was determined to be the most optimal by experiment [8].

Decompounding is the process where a compound word is taken and broken into its individual components. A compound word consists of two or more words that can also be used individually. These are glued together in the compound to form a new or related meaning. Both Dutch and German are languages in which compounding is common. Two different approaches have been tested. One that breaks compounded words in the shortest possible units (Decompounding S) and one that breaks them into the longest possible units (Decompounding L). Wordlists were used that were available for both the Dutch and German language. These are wordlists that are installed by default on modern UNIX systems; respectively *wdutch* that follows the 1996 Dutch spelling and *wngerman* that follows the 1998 German new orthography spelling. The decompounding engine adds the component parts of decompounded words to the index for each document. The same is done at query-time [3, 6].

Words can also be reduced to their stem form. This procedure removes inflection from words, as in changing 'walking' to 'walk'. It also transforms derivatives to their root, as in stemming 'national' to 'nation'. For finding the stem of a word the *Porter stemmer* was chosen. Both the unstemmed word form and the stemmed variant are added to the index and the query [1, 8, 14].

Remember that the documents are constructed based on several fields. These are: title, guideline, comments, keywords and the texts of the referring questions. While normally these fields are simply combined into one document, field weighting uses a different approach and assigns a weight to each individual field. For example: when a keyword in a query is found in a document title it has a bigger influence on the final document rank than when the same keyword appears in the comment section of that document. We call this *field weighting*. Suitable weights were determined by experiment. Weight pre-multiplication was done during the indexing phase. This method does not require extra calculations during query execution and ranking time.

Table 2: *Categorical clustering.*

Rank	1	2	3	4	5	6	7	8	9	10
Document	a	b	c	d	e	f	g	h	j	k
Category	2	2	2	2	5	4	3	7	6	2

Finally, *word relation expansion* considers all words in a document and all words in a query and finds synonyms for each of them. Of course those synonyms also in turn have synonyms, but only one level of depth is used. All that is done is adding extra synonym words at the end of each document and at the end of a query as well. Those synonyms are extracted from EuroWordNet. Note that the German WordNet is only about one third the size of the Dutch WordNet. This somewhat impairs performance for German [10, 23].

3.2 Ranking Techniques

For the basic ranking process initially mostly tried and tested techniques were chosen like *term frequency / inverse document frequency* ($tf \cdot idf$) combined with the *vector-space model* (vsm). We later experimented with *Okapi Best Match 25* (BM25). We chose for these techniques since they are well understood and provide a solid foundation for a production system [16, 17, 18, 20].

3.3 Post-Processing Techniques

One of the interesting properties of the database is that all questions are categorised. Implicitly each answer can also belong to several categories, since an answer can be referred to by one or more questions. Thus, each document belongs to one or more categories. This characteristic can be exploited. The idea behind this approach is to look at the representation of the categories within the result list after executing a query. Imagine that there is a list with ten results with categories as displayed in Table 2. The first four documents and the tenth document belong to category two (the simplifying assumption that each document belongs to only one category is made).

Since category two is so strongly represented in the result set, document k that is now in position ten might actually better be in position five. This intuition is precisely what *categorical clustering* tries to capture. If a category is strongly represented in the result list, with respect to the entire dataset, the documents that also fall in that category are pushed up. To achieve this first the representation strength of each category is determined based on the entire database. The formula is similar to *idf*, but applied to category counts:

$$icf\,(c) = \frac{\#D}{\#\{d \mid (d,c) \in D\}} \tag{1}$$

where D is the set of all documents in the entire database and their category. Finally, recalculating the scores is done as follows:

$$score_2\,(d) = score\,(d) \cdot \sum_{c \in C_d} icf\,(c) \tag{2}$$

where $score\,(d)$ is the original strength of document d and C_d is the set of *all* categories document d belongs to.

Besides categories, the MRSA corpus also contains links between answers. Each document contains a list of references to other documents that might also be relevant. To exploit this, we take the same approach as with categorical clustering: first a query is executed as usual and then the result list is examined and re-ranked. The references to other documents in the top N documents in the result list are collected. After that all documents in the list are reweighted based on the frequency in which they appear in this collection of referred documents.

Table 3: *Referential clustering.*

Rank	*Weight*		*Reference*
1	10	→	6
2	8	→	6
3	6	→	6
4	4	→	7
5	2	→	8
6	1	→	9

Let us look at the example in Table 3 which shows a result list consisting of six items. The sixth item with weight one falls just outside of the top five. Notice that document one, two and three all carry a reference to document number six, this is called a referential cluster. Reweighting proceeds by simply multiplying the score of document six by three which yields $1 \cdot 3 = 3$. Hence, the document in position five with score two is pushed down in the result list and replaced by the document in position six. Hence, a document in the result list which is indicated to be relevant by the top of the list has been pushed up. Of course, only the top N documents should have influence on pushing up other documents. A value of N of 5 was determined by experiment. In addition to this the frequency of appearance in the referring document list is dampened by the log function which results in more conservative multiplication behaviour (this is not applied in the example).

To formalise we first calculcate the frequency of appearance of a document d within the references:

$$frequency\,(d) = \#\{r \mid (r,d) \in R_5\} \tag{3}$$

where R_5 is the partial rank list consisting of the first *five* ranks r and their reference(s) d. R_5 can actually contain more than five pairs in case of multiple references at the same rank.

The actual reweighting now proceeds as follows:

$$score_2\,(d) = score\,(d) \cdot f\,(frequency\,(d)) \tag{4}$$

where $score\,(d)$ is the original weight of document d and f is a monotonous function. We used the non-linear: $f\,(x) = \log_2\,(\max\,(x+1,2)))$. The formula ensures that only documents that are referred to *multiple times* can affect the ranking. Rescoring is performed over the entire ranked list.

Latent Semantic Indexing (LSI) is a dimensionality reduction method capable of finding semantic relations. Unfortunately it increases recall and tends to decrease precision which is the opposite of what is desirable for the MRSA-QA system. It is also of less use when the corpus is homogeneous and is computationally expensive. A cheaper method with similar capabilities is *blind relevance feedback*. Each document in a result set contains many more words than only the words in the query. This can be exploited by adding the top M most frequently occurring terms in the top N documents

in the ranked list to the original query and re-executing the search with this newly expanded query. So, each time a query is posed to the system two passes are made: one to determine the initial ranked result list and extract the top terms, and then another with the expanded query. For the number of documents N a value of at most[2] three was chosen for selecting the M top terms. The value of M was set to five. An extra constraint was added so only terms of at least length four are used. This is to avoid selecting smaller non content-words, such as articles [11, 12].

4. EVALUATION

4.1 Sets & Metrics

For evaluation purposes several sets of queries were created by hand. Every set has the following characteristics: For each query the most relevant document is indicated. For each document in each of the databases there is at most one query in a single set, but there may also be no query. Nevertheless, most sets contain queries for nearly all answers in the associated database. The difference between the evaluation sets is in the word usage in the queries: each set contains a different 'wording' which attempts to retrieve the same answer. An example:

EXAMPLE 1. *Ways to retrieve document number 64, in the evaluation set, concerning 'how many times someone can acquire MRSA' (English translations shown in upright font).*

hoeveel keer MRSA
how many times MRSA

risicofactoren infecties
risk factors infections

hoe is het mogelijk om vaker MRSA te krijgen?
how is it possible to acquire MRSA multiple times?

hoe groot is de kans dat infectie vaker optreedt?
how big is the chance that the infection re-occurs?

The first two queries shown are keyword queries, while the last two are question queries. For the final evaluations eight sets for each domain / language combination were used, four with keyword-based queries and four with question-based queries. Note that not all sets cover all documents in the database. They were weighted respective to their size for computations involving averages and deviations. All combined over 6000 queries were executed for every evaluation run. The assumption is made that the queries in the sets are somewhat representative of the queries posed by 'real' users of the system. It is difficult to give any guarantees regarding this, especially since the evaluation sets are based on the answers that are present in the databases (they are based on the answers that the system *should be able* to provide). Real end-users might come up with radically different phrasings. Also, closed domain systems are usually faced with longer queries on average than open domain systems. Usually the average query length is about five terms per query. This trait is also present in the evaluation query sets, which gives a positive indication of their representativity [13].

When a query is executed the system generates a list of documents that are sorted according to their presumed relevance. For measuring how well the system performs it is necessary to define what is considered to be a good list of returned documents. Note that for any single query that is executed three types of document may be returned: zero or one most relevant document, zero or more other documents that are partially relevant and zero or more irrelevant documents. Armed with these concepts the definition of what the system should do is: The most relevant document should be at position 1 in the list. All other n partially relevant documents should populate position 2 up to $n + 1$. In the evaluation sets the most relevant document is indicated explicitly and the partially relevant documents are those that the most relevant document refers to.

For expressing the position of the most relevant document the Mean Reciprocal Rank (MRR) was used. Mean Average Precision (MAP) and Mean R-Precision (MRP) are used for measuring the presence of other relevant documents. They measure precision and recall, but are in fact highly correlated as they both estimate the area under the recall-precision curve. Hence, they should show similar results during evaluation. A non-rank related performance measure was also used, namely the time it takes for a single query to execute. Significance testing was performed using one-tailed paired unequal variance t-tests. We recognise that these tests make assumptions about the shape of the data which might not necessarily hold, but we believe that the resulting alpha values can still be compared [2, 4, 13].

4.2 Baseline

From here on forward an implementation of a technique will be referred to as an engine. The theoretical techniques as explained in this paper provide the blueprints for these practical implementations.

For evaluating the effectiveness of the techniques described previously, we need some sort of baseline to compare against. Six candidate baseline engines were developed. All of these engines use a basic bag-of-words approach. The difference is in the weighting method that is used. Note that as part of standard pre-processing all documents are stripped of any mark-up tags they may have. Also, words are lowercased and stripped of diacritical marks. Words consisting of only one character, or that do not include alphanumeric characters, are ignored. If a query is entered the documents in which at least one of the words in the query appears, which is determined by looking at the index, become part of the resulting set of documents. After this the frequencies of the words are used to assign a score to each document and turn the set of selected documents into a ranked list.

Roughly there are two approaches that have been evaluated. The first works by treating both query and documents as vectors also known as the *vector-space model*. There are three such vector-based baseline engines: Vector, Vector Log and Vector Normalised Log. The idea behind all of them is the same. They differ only in the variants of the function that they apply. All of them performed quite poorly in comparison with the other methods that were tried [11, 18].

The other approach is conceptually simpler and works by summing the scores of each document on each individual query term. Three variants of this were also tried, namely Additive, Additive Log and a self-developed approach. Statistical tests were performed to determine which of the six engines performed best. Based on the MRR the Additive and self-developed variants scored best. While the latter is faster than the normal additive method it was not chosen since it was not tested outside the MRSA corpus.

[2]Depends on the size of the result list.

Two variants of the Additive approach were used, one employing the basic $tf \cdot idf$ formula (shown in equation 5 where d is a document and w_k is a query term) and the other (at a later stage) employing Okapi BM25 [15, 16].

$$score\,(d) = \sum_{k=1}^{n} tf\,(d, w_k) \cdot idf\,(w_k) \tag{5}$$

4.3 Individual techniques

First, we need to select techniques based on their performance increment over the baseline. We report the average performance and deviation over four *keyword* evaluation sets and four *question* evaluation sets. The numbers in the tables shown are for the Dutch *professional* database using $tf \cdot idf$ scoring. Colours (shades) and arrows are used to indicate performance increment $\triangle$ (*green*) or decrement $\triangledown$ (*salmon*) over the baseline (*blue*). All numbers are rounded, so even if a performance number seems exactly the same as the baseline, it may still differ to the right of the last rounded digit. For the referential clustering engine the same data is used as for calculating the MAP and MRP scores which makes the increase in these values optimistic for this engine. Hence, those fields have been coloured ▲▼ (*pink*) in the various tables.

Table 4 shows that character n-gramming, decompounding and stemming positively affect the MRR. A similar pattern exists for the MAP and MRP shown in Table 5, but in contrast with the MRR the blind query expansion shows quite some improvement over the baseline here. Categorical clustering and word n-gramming decrease performance. Word relation expansion is fairly neutral and yields no convincing performance advantage. We believe this is due to the domain specific corpus and WordNet's generality [21].

Of course, besides the Dutch professional keyword sets there are also three other keyword evaluation sets. We describe performance of those sets in terms of differences with the tables:

- German/professional: Baseline keyword performance is quite a bit worse than for Dutch (MRR -0.06, MAP -0.05). The same holds for the question sets (MRR -0.08, MAP -0.05). The results show relatively similar performance patterns. However, German language appears to benefit more from both character n-gramming and stemming than Dutch, which is in line with findings of others [3, 8].
- Dutch/public: The overall performance is worse (MRR -0.10, MAP -0.01). Field weighting shows a slight performance increase which is the only exception with regard to the trends in performance on the Dutch professional dataset.
- German/public: Interestingly baseline performance on keywords is actually comparable to the Dutch professional dataset (MRR -0.01, MAP +0.03). Patterns are again similar, the exception being query expansion which performs a bit worse (MRR -0.02).

Performance on the Dutch question sets is shown in Table 6 and Table 7. Notice that the baseline performance on questions is much better than on keywords. This is probably caused in part by the inclusion of question-words (why, who, where, etcetera) in the question based queries.

Table 4: *Keywords: Reciprocal Rank & Timings (sec).*

	RR		*Time* $\times 10^{-3}$	
Engine	μ	σ	μ	σ
Baseline	0.51	0.40	7.1	5.5
Categorical clustering	$0.44^{\triangledown}$	$0.39^{\triangle}$	$10.0^{\triangledown}$	$8.4^{\triangledown}$
Character n-gramming	$0.54^{\triangle}$	$0.38^{\triangle}$	$25.4^{\triangledown}$	$11.3^{\triangledown}$
Decompounding large	$0.52^{\triangle}$	$0.39^{\triangle}$	$9.6^{\triangledown}$	$6.0^{\triangledown}$
Decompounding small	$0.55^{\triangle}$	$0.38^{\triangle}$	$13.0^{\triangledown}$	$7.0^{\triangledown}$
Field weighting	$0.49^{\triangledown}$	$0.40^{\triangledown}$	$7.4^{\triangledown}$	$6.8^{\triangledown}$
Query expansion	$0.47^{\triangledown}$	$0.39^{\triangle}$	$20.1^{\triangledown}$	$6.4^{\triangledown}$
Referential clustering	$0.51^{\triangledown}$	$0.39^{\triangle}$	$7.2^{\triangledown}$	$5.2^{\triangle}$
Stemming	$0.52^{\triangle}$	$0.39^{\triangle}$	$7.9^{\triangledown}$	$5.5^{\triangledown}$
Word n-gramming	$0.32^{\triangledown}$	$0.41^{\triangledown}$	$5.6^{\triangle}$	$5.0^{\triangle}$
Word relation expansion	$0.48^{\triangledown}$	$0.40^{\triangle}$	$9.7^{\triangledown}$	$6.9^{\triangledown}$

Table 5: *Keywords: Average Precision & R-Precision.*

	AP		*RP*	
Engine	μ	σ	μ	σ
Baseline	0.28	0.20	0.29	0.20
Categorical clustering	$0.26^{\triangledown}$	$0.20^{\triangle}$	$0.27^{\triangledown}$	$0.21^{\triangledown}$
Character n-gramming	$0.32^{\triangle}$	$0.19^{\triangle}$	$0.30^{\triangle}$	$0.19^{\triangle}$
Decompounding large	$0.31^{\triangle}$	$0.20^{\triangledown}$	$0.31^{\triangle}$	$0.20^{\triangle}$
Decompounding small	$0.33^{\triangle}$	$0.20^{\triangle}$	$0.31^{\triangle}$	$0.20^{\triangle}$
Field weighting	$0.29^{\triangledown}$	$0.21^{\triangledown}$	$0.29^{\triangle}$	$0.21^{\triangledown}$
Query expansion	$0.32^{\triangle}$	$0.21^{\triangledown}$	$0.30^{\triangle}$	$0.22^{\triangledown}$
Referential clustering	$0.33^{\blacktriangle}$	$0.23^{\blacktriangledown}$	$0.33^{\blacktriangle}$	$0.23^{\blacktriangledown}$
Stemming	$0.29^{\triangle}$	$0.20^{\triangledown}$	$0.30^{\triangle}$	$0.20^{\triangle}$
Word n-gramming	$0.15^{\triangledown}$	$0.20^{\triangle}$	$0.16^{\triangledown}$	$0.20^{\triangle}$
Word relation expansion	$0.29^{\triangle}$	$0.20^{\triangledown}$	$0.28^{\triangledown}$	$0.20^{\triangle}$

Positive increments are visible for character n-gramming, small decompounding and field weighting across all the metrics. Again, the performance differences on the other *question* evaluation datasets:

- German/professional: Similar performance as on keywords, baseline question performance is worse (MRR -0.06, MAP -0.05). Although the increase caused by stemming and character n-gramming is higher.
- Dutch/public: Shows a reduction in baseline performance (MRR -0.19, MAP -0.06). Relative to this character n-gramming yields a slightly higher improvement (MRR +0.05, MAP +0.04).
- German/public: Baseline performance is quite poor (MRR -0.22, MAP -0.09). The overall relative patterns are the same, but character n-gramming shows a slightly higher performance gain on par with Dutch public dataset.

The final choice is based on the significance of difference between MRR and MAP α values. Only methods that had at least one significant positive effect and no negative effect with respect to baseline on the professional dataset for both of these metrics and for both languages were selected. The final selection of five of the ten techniques is: character n-gramming, decompounding small, field weighting, referential clustering and stemming.

Table 6: *Questions: Reciprocal Rank & Timings (sec).*

	RR		*Time* $\times 10^{-3}$	
Engine	μ	σ	μ	σ
Baseline	0.60	0.39	17.2	7.9
Categorical clustering	0.47^{∇}	0.41^{∇}	23.4^{∇}	10.2^{∇}
Character n-gramming	0.63^{Δ}	0.37^{Δ}	61.4^{∇}	28.6^{∇}
Decompounding large	0.59^{∇}	0.39^{Δ}	30.0^{∇}	13.8^{∇}
Decompounding small	0.61^{Δ}	0.38^{Δ}	36.0^{∇}	16.2^{∇}
Field weighting	0.61^{Δ}	0.40^{∇}	41.1^{∇}	21.5^{∇}
Query expansion	0.55^{∇}	0.38^{Δ}	40.1^{∇}	13.3^{∇}
Referential clustering	0.58^{∇}	0.38^{Δ}	17.2^{Δ}	7.8^{Δ}
Stemming	0.60^{∇}	0.39^{Δ}	19.5^{∇}	9.0^{∇}
Word n-gramming	0.51^{∇}	0.41^{∇}	8.3^{Δ}	5.9^{Δ}
Word relation expansion	0.49^{∇}	0.40^{∇}	46.9^{∇}	28.0^{∇}

Table 7: *Questions: Average Precision & R-Precision.*

	AP		*RP*	
Engine	μ	σ	μ	σ
Baseline	0.34	0.20	0.33	0.20
Categorical clustering	0.28^{∇}	0.21^{∇}	0.26^{∇}	0.22^{∇}
Character n-gramming	0.36^{Δ}	0.20^{Δ}	0.34^{Δ}	0.19^{Δ}
Decompounding large	0.34^{∇}	0.20^{∇}	0.33^{∇}	0.21^{∇}
Decompounding small	0.35^{Δ}	0.20^{Δ}	0.33^{Δ}	0.20^{Δ}
Field weighting	0.35^{Δ}	0.20^{Δ}	0.33^{Δ}	0.21^{∇}
Query expansion	0.36^{Δ}	0.21^{∇}	0.33^{Δ}	0.21^{∇}
Referential clustering	$0.44^{\blacktriangle}$	$0.25^{\blacktriangledown}$	$0.41^{\blacktriangle}$	$0.24^{\blacktriangledown}$
Stemming	0.34^{∇}	0.20^{Δ}	0.33^{Δ}	0.20^{Δ}
Word n-gramming	0.25^{∇}	0.20^{Δ}	0.26^{∇}	0.19^{Δ}
Word relation expansion	0.29^{∇}	0.19^{Δ}	0.27^{∇}	0.20^{Δ}

4.4 Optimal value of n

A value of five was chosen for character n-gramming. This choice was initially based on the average length of the words in the professional Dutch and German datasets. Since character n-gramming is quite a fundamental difference with normal bag-of-words indexing, it is important to show that this is also the beste value for n. We tested several values of n for both the Dutch and German professional datasets. These tests were conducted on top of the additive $tf \cdot idf$ baseline that was chosen earlier.

Figure 2 shows how the value of n, ranging from two to eight, affects the MRR and MAP. For keyword queries 4-grams are slightly more optimal for Dutch with respect to 5-grams according to the MRR. Nevertheless, for German keyword queries this makes no difference. For question queries the picture is a bit different: the optimal value of n appears to be six for Dutch, while for German the performance tops-off after an n of five. A similar trend can be seen for the other measures. The differences however are not very large. Based on the trend-line the optimal value of n is four or five. This is the same finding as McNamee & Mayfield who show that these values give optimal performance for most European languages [8, 12].

An other measure is the retrieval time. Due to their length, this is significantly longer for question-based queries than for keyword-based ones. The lower the value of n, the larger the inverted index. Hence, higher values of n are preferred simply because of their retrieval speed advantage. But the speed difference between subsequent values of n becomes smaller as n increases: for 7 and 8-grams the difference is minimal. Choosing between 4 and 5-grams, based on speed

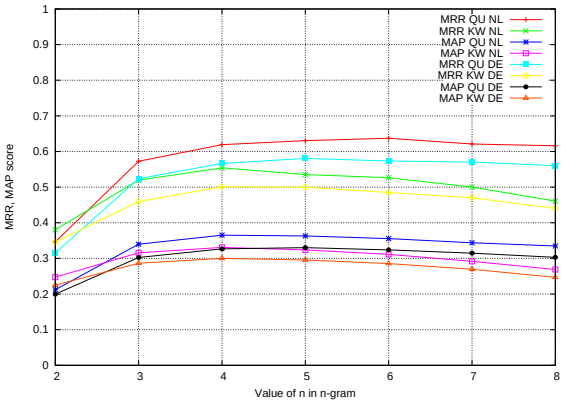


Figure 2: *MRR and MAP scores for n-grams on professional keyword (KW) and question (QU) evaluation sets for Dutch (NL) and German (DE).*

alone, 5-grams are preferred. McNamee & Mayfield report a tenfold speed penalty when using n-grams, but in our tests 5-grams are 'only' four times slower than the baseline approach. This confirms their own suspicions regarding the fact that the increase in processing time while using n-grams is an artefact of their implementation [12].

We eventually settled on a value for n of five, since it only slightly degrades performance on the Dutch MRR for keyword-based queries and has a mild positive effect for all question-based queries. 5-grams also have a speed performance advantage over 4-grams ($\sim$ 25% for question queries and $\sim$ 15% for keyword queries). It is safe to confirm McNamee & Mayfield's conclusion that higher values of n are beneficial for time-wise performance. It can be generally stated that: 'For n-grams and n+1-grams, the n+1-grams are preferred speed-wise when there exists no significant difference in ranking performance between the n-grams and n+1-grams'.

4.5 Combined techniques

Knowing the performance of individual techniques, we can combine them to improve performance. Abbreviations are used for the five previously selected techniques in this section: (C)haracter n-gramming, (D)ecompounding Small, (F)ield weighting, (R)eferential Clustering and (S)temming.

As can be seen in Table 8 and Table 9, combining character n-gramming with field weighting yields poor results for the Dutch MRR's. Hence, field weighting was dropped. The next combination that was tried was character n-gramming plus stemming, this does show improvement, especially for keyword based queries. After this small word decompounding was stacked on n-gramming and stemming. While this does not degrade retrieval performance overall, it does not increase it either. As a downside decompounding adds quite some overhead to the processing time (factor 1.5) and (not visible in the tables here) also adds significant processing time while indexing. Based on this and the lack of any significant performance increase, small decompounding was dropped. The last addition that was made is that of referential clustering. This does not really affect the MRR, but

Table 8: *Keywords: Combined performance on Dutch KW sets.*

	RR		*AP*		*Time* $\times 10^{-3}$	
Engine	μ	σ	μ	σ	μ	σ
Baseline	0.51	0.40	0.28	0.20	7.1	5.5
CF	0.47^{∇}	0.39^{Δ}	0.29^{Δ}	0.18^{Δ}	62.5^{∇}	28.6^{∇}
CS	0.55^{Δ}	0.38^{Δ}	0.34^{Δ}	0.19^{Δ}	33.1^{∇}	15.0^{∇}
CSD	0.56^{Δ}	0.38^{Δ}	0.34^{Δ}	0.19^{Δ}	55.2^{∇}	26.6^{∇}
CSR	0.55^{Δ}	0.38^{Δ}	0.40^{Δ}	0.23^{∇}	36.5^{∇}	16.2^{∇}

Table 9: *Questions: Combined performance on Dutch QU sets.*

	RR		*AP*		*Time* $\times 10^{-3}$	
Engine	μ	σ	μ	σ	μ	σ
Baseline	0.60	0.39	0.34	0.20	17.2	7.9
CF	0.55^{∇}	0.38^{Δ}	0.32^{∇}	0.18^{Δ}	162.6^{∇}	78.8^{∇}
CS	0.63^{Δ}	0.37^{Δ}	0.37^{Δ}	0.20^{Δ}	82.8^{∇}	40.5^{∇}
CSD	0.64^{Δ}	0.37^{Δ}	0.37^{Δ}	0.20^{Δ}	126.3^{∇}	64.0^{∇}
CSR	0.62^{Δ}	0.37^{Δ}	0.45^{Δ}	0.23^{∇}	85.3^{∇}	41.3^{∇}

it does increase the MAP. While this is to be expected, since the evaluation calculation uses the same data as is used for referential clustering, it is still an interesting addition. Since referential knowledge is available there is nothing against using it, especially because of the low processing overhead and the fact that it does not negatively affect the MRR.

The final engine consists of character n-gramming, stemming and referential clustering combined. We made some final optimizations to the implementation using acceleration tables and code improvements leading to a higher processing speed especially for question based queries ($\sim 2\times$ faster).

An important question: is the difference between the baseline engine and this new combined engine statistically significant? To answer this question we look only at the professional database and its evaluation results. Alpha values are shown in Table 10. With respect to the MRR we can conclude that the improvement of the CSR engine over the baseline is only significant for German keyword queries, even though the results on the other three sets are close to weak significance. However, the improvement to the MAP is significant. Even without the referential clustering all have $\alpha \leq 0.026$. Increasing the MRR is more difficult due to the fact that, even with different techniques applied, it has a high standard deviation.

Apart from $tf \cdot idf$ we further experimented with improving the performance of the final combined engine using Okapi BM25. For the public dataset using BM25 slightly worsened performance on keyword and question sets. However, on the professional dataset it did not adversely affect keyword performance and actually improved performance on questions: MRR +0.06 and MAP/MRP +0.04 with no change in the deviations. Hence, the decision was made to use the $tf \cdot idf$ based combined engine for querying the public databases and the BM25 variant for the professional ones.

Table 10: *CSR's significance of improvement over baseline.*

	Keyword Sets α		*Question Sets* α	
Language	*MRR*	*MAP*	*MRR*	*MAP*
Dutch	0.119	<0.001	0.251	0.001
German	0.039	0.001	0.112	<0.001

5. CONCLUSION

A working question-answering system for the MRSA domain has been developed based on research into Information Retrieval and Natural Language Processing techniques. Not all of the initially selected techniques work well on the dataset: only half of them show favourable performance increments. Appropriate techniques are highly dependent on both the size and content of the corpus. Curiously the relatively obscure character n-gram technique scored very well. This is presumably due to the fact that it properly captures local word sequence relations. The value chosen for n has been shown to be optimal. While some other techniques showed initial promise as well, combining them led to a decline in performance, which shows that techniques that improve performance individually may conflict with other such techniques.

The evaluation results show the need to base oneself not on only one (favourable) evaluation statistic, but on multiple ones to give a more accurate picture of the performance of the various techniques and the system as a whole. It is also important to look not only at the mean performance, but also at the stability using the *deviation*. The MRR has been shown to be quite hard to increase, which is largely due to the high deviation for this statistic.

Interestingly all the applied techniques and even the baseline yield better retrieval performance for question-based queries then keyword-based queries. This supports the conclusion that in general the retrieval task is easier to perform on question-based queries even without using techniques specifically geared towards these type of queries.

The initial research question was: 'How can the user be provided with the most appropriate answer(s) for his or her query within the restricted MRSA domain given the available structured MRSA corpus?'. This paper answers that question by providing a range of techniques that apply specifically to this corpus. The final selection of combined techniques is a blend between Information Retrieval (IR) (additive $tf \cdot idf$ and BM25), Natural Language Processing (NLP) (stemming, character n-grams) and novel usage of extra information present in the corpus (referential clustering). Several techniques might apply well to other corpora, but only decrease performance on this one. Also, there are approaches that increase the average precision, but decrease the reciprocal rank as side effect. This leads to the conclusion that the techniques to be chosen strongly depend both on the corpus, the query formulation that is used and the statistic for which one wants to optimise.

The evaluation results provide an interesting hint on the techniques that could be tried for IR use on other Dutch and German corpora. Especially character n-gramming shows much promise for broad application and is not commonly used for retrieval purposes. Also, when a corpus exhibits structural information it is highly recommended to investigate how this information can be used to increase retrieval performance.

Reasoning from the vantage point of the user is very important for IR systems that need to be used in practice, but this is unfortunately frequently neglected in favour of scientific performance measurements alone. While this paper has focused only on evaluating the system in a semi-automated fashion, real user investigations have also been carried out, leading to the current user-centered design of the MRSA web-portal [22].

6. FUTURE WORK

There are many other techniques that might be useful and could be tried on the MRSA corpus using the evaluation framework that was built for this research. We believe application of more sophisticated techniques to be a good direction for future research. This includes *explicit relevance feedback, ontology* and *word proximity* approaches, *s-grams*, and use of *syllables* instead of *n*-grams. *Part-of-speech tagging* could be used for example to filter out function words. An investigation into the EuroWordNet coverage of the medical domain and the word relations therein may lead to better results for word relation expansion. A functional enhancement that could be made is setting up the system as a dialog system that helps the user refine the posed queries. This could also be tied to a specific user profile for learning from and adapting the system to the user [6, 9, 11, 19].

Acknowledgements

We wish to thank Djoerd Hiemstra and Maarten Fokkinga for their support. Useful feedback and suggestions on prior incarnations of this paper were also provided by Danny Oude Bos, Desislava Dimitrova, Maarten Eykelhoff, Marco Gerards, Hendri Hondorp, Marco Pasch, Boris van Schooten, Rianne Tigelaar and Ruben Wassink.

This study was financially supported by the EUREGIO MRSA-net Twente/Münsterland project.

7. REFERENCES

[1] Ahlgren, P., and Kekäläinen, J. Swedish full text retrieval: Effectiveness of different combinations of indexing strategies with query terms. *Information Retrieval 9*, 6 (2006), 681–697.

[2] Aslam, J. A., Yilmaz, E., and Pavlu, V. A Geometric Interpretation of R-precision and Its Correlation with Average Precision. In *Proceedings of SIGIR* (Salvador, BR, August 2005).

[3] Braschler, M., and Ripplinger, B. How Effective is Stemming and Decompounding for German Text Retrieval? *Information Retrieval 7* (2004), 291–361.

[4] Buckley, C., and Voorhees, E. M. Evaluating Evaluation Measure Stability. In *Proceedings of SIGIR* (Athens, GR, July 2000).

[5] Burke, R. D., Hammond, K. J., Kulyukin, V., Lytinen, S. L., Tomuro, N., and Schoenberg, S. Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System. Tech. Rep. TR-97-05, University of Chicago, 1997.

[6] Chen, A., and Gey, F. C. Multilingual Information Retrieval Using Machine Translation, Relevance Feedback and Decompounding. *Information Retrieval 7* (2004), 149–182.

[7] Hirschman, L., and Gaizauskas, R. Natural language question answering: the view from here. *Natural Language Engineering 7*, 4 (2001), 275–300.

[8] Hollink, V., Kamps, J., Monz, C., and de Rijke, M. Monolingual Document Retrieval for European Languages. *Information Retrieval 7* (2003), 33–52.

[9] Järvelin, A., Järvelin, A., and Järvelin, K. s-grams: Defining generalized n-grams for information retrieval. *Information Processing and Management 43* (2007), 1005–1019.

[10] Loerch, U., and Guesgen, H. Constructing an Intelligent Query Answering System. In *Proceedings of ANNES* (Dunedin, NZ, November 2001).

[11] Manning, C., and Schütze, H. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.

[12] McNamee, P., and Mayfield, J. Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval 7* (2004), 73–97.

[13] Mishne, G., and de Rijke, M. Boosting Web Retrieval through Query Operations. In *Proceedings of ECIR* (Santiago de Compostela, ES, July 2005).

[14] Porter, M. F. An Algorithm for Suffix Stripping. *Program 14*, 3 (1980), 130–337.

[15] Robertson, S. E. Understanding Inverse Document Frequency: On theoretical arguments for IDF. *Journal of Documentation 60*, 5 (2004), 503–520.

[16] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. Okapi at TREC-3. In *Proceedings of TREC* (Gaithersburg, MA, USA, November 1994).

[17] Salton, G., and Buckley, C. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management 24*, 5 (1988), 513–523.

[18] Salton, G., Wong, A., and Yang, C. A Vector-Space Model for Automatic Indexing. *Communications of the ACM 18*, 11 (1975), 613–620.

[19] van Schooten, B., and op den Akker, R. Follow-up utterances in QA dialogue. *Traitement Automatique des Langues 46*, 3 (2007), 181–206.

[20] Spärck-Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28*, 1 (1972), 11–21.

[21] Srinivasan, D. P. . G. R. . S. Passage Scoring for Question answering via Bayesian inference on lexical relations. In *Proceedings of TREC* (Gaithersburg, MA, USA, November 2003).

[22] Verhoeven, F., Hendrix, R. M., Daniels-Haardt, I., Friedrich, A. W., Steehouder, M. F., and van Gemert-Pijnen, J. E. The development of a web-based information tool for cross-border prevention and control of Methicillin Resistant Staphylococcus Aureus. *International Journal of Infection Control 4*, 1 (2008).

[23] Vossen, P. Introduction to eurowordnet. *Computers and Humanities 32* (1998), 73–89.

Learning to Rank Answers to Why-Questions

Suzan Verberne* s.verberne@let.ru.nl
Stephan Raaijmakers† stephan.raaijmakers@tno.nl
Daphne Theijssen* d.theijssen@let.ru.nl
Lou Boves* l.boves@let.ru.nl

ABSTRACT

The goal of the current research project is to develop a question answering system for answering *why*-questions (*why*-QA). Our system is a pipeline consisting of an off-the-shelf retrieval module followed by an answer re-ranking module. In this paper, we aim at improving the ranking performance of our system by finding the optimal approach to learning to rank. More specifically, we try to find the optimal ranking function to be applied to the set of candidate answers in the re-ranking module. We experiment with a number of machine learning algorithms (i.e. genetic algorithms, logistic regression and SVM), with different cost functions.

We find that a learning to rank approach using either a regression technique or a genetic algorithm that optimizes for MRR leads to a significant improvement over the TF-IDF baseline. We reach an MRR of 0.341 with a success@10 score of 58.82%. We also see that, as opposed to logistic regression and genetic algorithms, SVM is not suitable for the current data representation. After extensive experiments with SVMs, we still reach scores that are below baseline.

In future work, we will investigate in more detail the limitations of our re-ranking approach: which set of questions cannot be answered in the current system set-up and why?

General Terms

Question Answering, *Why*-Questions, Learning to Rank

1. INTRODUCTION

The goal of the current research project is to develop a question answering system for answering *why*-questions (*why*-QA). In a QA system, *why*-questions need a different approach from factoid questions since their answers are explanations that cannot be stated in a single phrase. Answers to *why*-questions tend to be at least one sentence and at most one paragraph in length [17]. Therefore, passage retrieval (as opposed to named entity retrieval, which is generally used for factoid QA) appears to be a suitable approach to *why*-QA.

In previous work, we have developed a passage retrieval system for *why*-QA [18]. This pipeline system consists of an off-the-shelf retrieval engine (Lemur[1]), extended with a re-ranking module that is specifically tuned for ranking the answers to *why*-questions. In the re-ranking module, a set of features is extracted from the question and each of the candidate answers retrieved by Lemur. The values of these features are combined in a ranking function that is used for re-ordering the set of candidate answers.

The task of finding the optimal ranking function for a specific information retrieval task is referred to as 'learning to rank' in the literature [12]. Until now, we have mainly focused on improving the ranking performance of our system by adapting and expanding the feature set used for re-ranking [18]. This has led to a set of 37, mostly linguistic, features.

In the current paper, we aim at improving the ranking performance of our system by finding the optimal approach to learning to rank. More specifically, we try to find the optimal ranking function to be applied to the set of candidate answers in the re-ranking module. We vary our experimental settings in two dimensions: the machine learning techniques (genetic algorithms, logistic regression and support vector machines), and the cost function. In all experimental settings, we keep the set of 37 features that we found to be relevant in previous work.

This paper is organized as follows: in Section 2, we discuss related work on approaches to learning to rank. In Section 3 we describe the resources that we use for our experiments and we specify the characteristics of the machine learning data. Section 4 defines the machine learning problem that we consider in our learning to rank experiments. In Section 5 and 6 we describe the experiments that we conducted and the results we obtained. The results are discussed in Section 7. Section 8 contains our conclusions.

2. RELATED WORK

*Dept. of Linguistics, Radboud University Nijmegen
†TNO Information and Communication Technology, Delft

[1]See http://www.lemurproject.org/

As explained in Section 1, we vary our experimental settings in two dimensions: the machine learning techniques that we use and the cost function that we implement. Therefore, we discuss related work in two subsections.

2.1 Machine learning techniques for learning to rank

Most approaches to learning to rank consider the problem as a case of supervised learning. The training set is a matrix of feature vectors for a set of instances (the items to be ranked). Each item is assigned a label representing its relevance ground truth. A supervised learning problem can be solved by regression and classification techniques. In [12], many approaches to learning to rank are discussed.

In previous work, we used a genetic algorithm for finding the optimal ranking function. Genetic algorithms are devised for sampling (finding an optimum in) a very large data space. The definition of 'optimum' here is defined by the so-called fitness function in the genetic algorithm. Genetic algorithms have been applied to learning to rank problems and other retrieval optimization problems by several researchers in the field [16, 6, 15]. The approach presented in [16] resembles our approach: it defines the learning problem as the search for the optimal weight vector for a given feature vector.

2.2 Cost functions for learning to rank

One of the advantages of genetic algorithms is that the cost function (fitness function) is user-defined[2]. In [6], a number of fitness functions that are derived from evaluation measures (such as average precision) are compared for their effectiveness.

An important aspect of the cost function in learning to rank problems is the definition of the ordering principle: items can be placed on an ordinal scale based on a score that is assigned to them[3] or they can be ordered relative to other items in the list by defining for each possible pair of items which of the two is more relevant. The latter learning principle is called 'pairwise preference learning', and was introduced by Joachims [9], who created the learning algorithm Ranking SVM based on this principle. In pairwise preference learning, the measure that is optimized is Kendall Tau:

$$\tau = (P - Q)/(P + Q) \quad (1)$$

in which P is the number of concordant item pairs (the two items are ordered correctly) and Q is the number of discordant item pairs (the two items are ordered incorrectly). Pairwise preference learning has been studied in more detail by Furnkranz and Hullermeier [7] and applied to several ranking problems such as combining rankings from multiple retrieval systems by Carterette and Petkova [3].

3. DATA AND SYSTEM SET-UP

3.1 Resources

For our experiments, we used the Wikipedia INEX corpus [5]. This corpus consists of all 659,388 articles extracted from the online Wikipedia in the summer of 2006, converted to XML format.

Before indexing the corpus, we segmented all Wikipedia documents in passages. We decided on using a semi-fixed passage size of 500 to 600 characters (excluding all XML markup) with an overflow to 800 for the purpose of completing sentences[4]. We create passage overlap by starting each new passage at a paragraph or sentence boundary halfway the previous passage[5]. For Wikipedia articles that contain less than 500 characters in total, we included the complete text as one passage.

Our segmentation process gives an index of 6,365,890 passages with an average length of 429 characters (standard deviation 194) per passage[6]. We separately saved the document title and section heading as metadata for each passage.

For development and testing purposes, we exploited the Webclopedia question set by Hovy et al. [8]. This set contains questions that were asked to the online QA system *answers.com*. Of these questions, 805 (5% of the total set) are *why*-questions. For 700 randomly selected *why*-questions from this set, we manually searched for an answer in the Wikipedia XML corpus, keeping the remaining questions for future test purposes. 187 questions have at least one answer in the corpus. Extraction of one relevant answer for each of these questions resulted in a set of 187 *why*-questions and their reference answer. Let us give three examples to illustrate the type of data we are working with:

1. "Why do most cereals crackle when you add milk?" — "They are made of a sugary rice mixture which is shaped into the form of rice kernels and toasted. These kernels bubble and rise in a manner which forms very thin walls. When the cereal is exposed to milk or juices, these walls tend to collapse suddenly, creating the famous 'Snap, crackle and pop' sounds."

2. "Why didn't Socrates leave Athens after he was convicted?" — "Socrates considered it hypocrisy to escape the prison: he had knowingly agreed to live under the city's laws, and this meant the possibility of being judged guilty of crimes by a large jury."

3. "Why was cobalt named cobalt?" — "The word cobalt comes from the German kobalt or kobold, meaning evil spirit, the metal being so called by miners, because it was poisonous and troublesome (it polluted and degraded the other mined elements, like nickel)."

[2]This is not unique for genetic algorithms but it is one of the most typical characteristics of genetic algorithms.

[3]This score generally is the probability, assigned by a classification or regression model, that the item has either of the two labels [2].

[4]We assume that answer passages ending in an unfinished sentence are undesirable. However, if the hard maximum of 800 characters is reached, the passage is cut off between two words to prevent non-sentence contexts like tables to result in extremely long passages.

[5]Other work on passage retrieval for QA [10] shows that better retrieval results are achieved with fixed-sized, partly overlapping passages than with structure-based, disjoint passages (e.g. <p>-items, which are very variable in length).

[6]The average length is smaller than the predefined minimum length of 500 characters because after clean-up a significant number of articles is shorter than 500 characters.

In order to be able to do fast evaluation without elaborate manual assessments, we manually created one answer pattern for each of the questions in our set. The answer pattern is a regular expression that defines which of the retrieved passages are considered a relevant answer to the input question. In their original versions, the answer patterns were directly based on the corresponding reference answer, but in the course of the development and evaluation process, we extended the patterns in order to cover as much as possible of the variants of the reference answer that occur in the Wikipedia corpus. By following this iterative process, we prevented to miss relevant answers.

For example, for question 2 above, we developed the following answer pattern based on two variants of the correct answer that occur in the corpus: */(Socrates.* opportunity.* escape.* Athens.* considered.* hypocrisy | leave.* run.* away.* community.* reputation)/*[7] If a candidate answer matches the answer pattern then this answer is marked relevant, otherwise it is marked irrelevant. This evaluation method means that we defined relevance as a binary variable: an answer passage is either relevant or not.

3.2 System set-up

As we briefly mentioned in Section 1, our system for *why*-QA consists of three pipelined modules: (1) a question processing module that transforms the input question to a query by removing stop words and punctuation; (2) an off-the-shelf retrieval module that retrieves and ranks passages of text that share content with the input query; and (3) a re-ranking module that re-ranks the retrieved passages using features extracted from the question and each of the candidate answers. We aim to find the optimal ranking function to be applied in the re-ranking module. Thus, for our learning to rank experiments, we used the output of the retrieval module (2).

In the retrieval module, we used Lemur to retrieve 150[8] answers per question and rank them using TF-IDF as it has been built in in Lemur[9]. This gave us a set of 187 questions with 150 candidate answers per question, with for each pair of a question and a candidate answer a TF-IDF score. For re-ranking, feature values needed to be extracted from each of these 28,050 ($187 * 150$) question-answer pairs.

3.3 Feature extraction

From earlier work [18], we compiled a set of 37 features that are summarized in Table 1. We parsed the questions with the Pelican parser[10] and the candidate answers with the Charniak parser. Then we used a Perl script for extracting all feature values from the question, the answer candidate and both their parse trees.

Each feature represents the similarity between two item sets: a set of question items (for example: all question NPs, or the question subject) and a set of answer items (for example: all answer words, or all subjects in the answer). The value that is assigned to a feature is a function of the similarity between these two sets. For determining this similarity, we use a statistic derived from the Jaccard index that is adapted for duplicate terms in either of the two sets. For a set of question word tokens Q, a set of question word types Q', a set of answer word tokens A and a set of answer word types A', the similarity S between Q and A is defined as:

$$S(Q, A) = \frac{|Q \cap A'| + |A \cap Q'|}{|Q \cup A|} \quad (2)$$

3.3.1 Description of the features

Syntactic features and Wordnet expansion features. Details on the syntactic features and the WordNet expansion features can be found in Verberne et al. 2008 [18]. The features that deserve some extra attention here, are the features related to question focus (e.g. overlap between the question focus and the title of the answer document). We introduced the term question focus in analogy to linguistically motivated approaches to factoid QA for the topic of the question ("What is the question about?"). We defined three rules for determining the focus of a *why*-question: If the subject is semantically poor, the question focus is the (verbal or nominal) predicate: "Why do people sneeze?". In case of etymology questions, the focus is the subject complement of the passive sentence: "Why are chicken wings called Buffalo wings?". In all other cases, the focus is the syntactic subject of the question, e.g. "Why are flamingos pink?" [18].

Cue word feature. The cue word feature is the overlap between a fixed set of explanatory cue words and the set of answer words. We found the cue words in a way that is commonly used for finding answer cues: we queried the key answer words to the most frequent *why*-question on the web ("blue sky rayleigh scattering" for "Why is the sky blue?") to the MSN Search engine[11] and crawled the first 250 answer fragments that are retrieved by the engine. From these, we manually extracted all phrases that introduce the explanation. This led to 47 cue words/phrases such as *because*, *as a result of*, *which explains why*, etc.

Document structure features. The six document structure features cover information about the document context of a candidate answer passage: overlap between the question and the title of the Wikipedia document, overlap between the question and the title of the section in which the candidate answer occurs, and the relative position of the candidate answer in the document.

WordNet Relatedness feature. We define the relatedness between a question and an answer as the weighted average of the relatedness between each of the question words and each of the answer words:

$$REL(Q, A) = \frac{\sum_{q=1}^{m} \sum_{a=1}^{n} REL(w_q, w_a)}{m} \quad (3)$$

in which Q, A is the question-answer pair under consideration, w_q represents the question words, w_a the answer words, and m is the number of question words. As measure of relatedness, we choose the Lesk measure, which incorporates

[7] Note that the vertical bar separates the two alternatives.
[8] We experimented with a higher number of answer candidates but coverage was hardly improved when increasing this number to 500.
[9] In previous work [10], we experimented with other ranking models and TF-IDF came out as the best.
[10] See http://lands.let.ru.nl/projects/pelican/
[11] http://www.live.com

Table 1: Set of 37 features used in our re-ranking module

TF-IDF	The score that is assigned to a candidate answer by Lemur/TF-IDF in the retrieval module
14 Syntactic feats	Overlap between question constituents (e.g. subject, verb, question focus) and answer words
14 WordNet expansion feat	Overlap between the WordNet synsets of syntactic question constituents and answer words
1 Cue word feat	Overlap between candidate answer and a pre-defined set of explanatory cue words
6 Document structure feats	Overlap between question (focus) words and document title and section heading
1 WordNet Relatedness feat	Relatedness between question and answer according to the WordNet similarity tool [13]

information from WordNet glosses. It finds overlaps between the glosses of two concepts, also if they belong to different word classes [13].

3.3.2 Resulting feature vectors

Feature extraction led to a vector consisting of 37 feature values for each of the 28,050 items in the data set. We experiment with two types of normalization: L2 normalization over all feature values per item ('horizontal normalization') and L1 normalization over all values for one feature, grouped per question ('vertical normalization'). Each item (representing one question-answer pair) was automatically labeled '1' if the candidate answer matches the answer pattern for the question and '0' if it does not. In total, 295 (1%) items in our set were labeled '1' and the rest was labeled '0'.

4. THE LEARNING PROBLEM

Based on the goal of our work (see Section 1) and the data we work with (see Section 3), we can identify the following characteristics of the learning to rank problem that we aim to solve in this paper:

- We aim at developing a system for answering *why*-questions. In the development phase, we use a set of 187 *why*-questions that have been asked to an online QA system. The system that we build should be generalizable to new *why*-questions; it should not depend on a database of previously answered questions. For the evaluation set-up of our experiments, this means that we must split the training and test collections in such a way that are all answers to the same question occur in either the training set or the test set.
- In our data collection, we have much more negative than positive instances (99% has value '0'). This class imbalance means that the baseline for classification tasks on these data is extremely high: if a classifier would classify all instances as '0', then accuracy would be 99%. This is not desirable because the evaluation of the results is based on QA evaluation measures (see Section 5.2) and without positive instances in the output, the values of these measures will be zero.
- The previous point would suggest an approach based on ranking optimization, like it is performed by Ranking SVM [9]. However, Ranking SVM expects a ranked ground truth, i.e. multi-level evaluation (as opposed to binary labels 0 and 1). Since we defined the relevance of the answers as a binary variable, our learning problem seems more suited for classification than for ranking optimization.
- We use a set of features between which complex relations exist. Some of our features are correlated and others even depend on each other. For example, when the overlap between the main verb in the question and the verbs in the answer passage is > 0 then the overlap between the main verb in the question and all words in the answer passage is automatically > 0 as well. For this paper, we did not calculate the correlations and dependencies between all pairs of 37 features, so we cannot completely oversee the complexity of the feature set. Complex feature relations may cause challenges when using linear classification algorithms.

5. EXPERIMENTS

5.1 Baseline

As baseline we use the system setting in which the answers are retrieved and ranked by Lemur/TF-IDF, without application of the re-ranking module. Thus, in the baseline setting, the answers are ranked according to the single feature value TF-IDF.

5.2 Evaluation set-up

After labeling each of the instances with 0 or 1 with use of the answer patterns (see Section 3.1), we count the questions that have at least one relevant answer in the top n ($n = 10, 150$) of the results. This number divided by the total number of questions in our test collection gives the measure *success*@n. For the highest ranked relevant answer per question, we determine the reciprocal rank (RR). If there is no relevant answer retrieved by the system at $n = 150$, the RR is 0. Over all questions, we calculate the mean RR: MRR@150.

In the learning to rank stage, we perform 5-fold cross validation on the question set. We keep the 150 answers to each question together in one fold so that we do not train and test on the answers to the same question.

5.3 Learning algorithms and optimization functions

In this section, we give an overview of the learning algorithms and cost functions that we use for our experiments, and how we apply them to our learning problem. In each of the settings, we use the 37-feature set that we described in Section 3.3.

5.3.1 Genetic algorithm

As we pointed out in Section 2, genetic algorithms have the advantage that the cost function ('fitness function') is user-defined. This means that genetic algorithms allow us to experiment with different cost functions and to optimize directly for ranking performance (MRR or some related measure). Our aim when training the genetic algorithm is to find the optimal weight vector for our feature vector of 37

feature values. As weights, we use the integers 0 to 10. In terms of the genetic algorithm, each possible weight vector is an individual. For each individual that is generated by the algorithm, our fitness function linearly multiplies the weight vector with the feature vectors for all items in the training set. This leads to a new score for each item.

We experiment with two fitness functions in the genetic algorithm:

1. MRR. The fitness function converts new item scores to ranks by simply sorting them per question, and then calculates MRR over the complete training set. By adapting ('evolving') the weight vector over a number of generations[12], the genetic algorithm optimizes MRR for the training set.
2. Pairwise preference learning. We implement a fitness function that optimizes Kendall Tau (see Equation 1 in Section 2). In the fitness function, all pairs of one positive (1) and one negative (0) item are selected from the training set and their newly calculated scores are compared. If the positive item has a higher score than the negative item, the pair is concordant — otherwise it is discordant. From the counts for concordant and discordant pairs, the fitness function calculates τ.

5.3.2 Logistic regression

We use the *lrm* function from the Design package in R[13] for training and evaluating models based on logistic regression. Using a set of input variables (features) logistic regression establishes a function that determines the log of the odds that the item is relevant (has label '1'). The log odds are defined as:

$$\ln odds(rel_i = 1) = \frac{e^{P(rel_i=1)}}{1 + e^{P(rel_i=1)}}, \quad (4)$$

in which $P(rel_i = 1)$ is the probability that item i is relevant.

The regression function that outputs the log odds is defined as follows:

$$\alpha + \vec{\beta_k}\vec{V_{ik}}, \quad (5)$$

in which α is the intercept. $\vec{\beta_k}\vec{V_{ik}}$ are the weights β and values V_i of the features k. The optimal values for α and β_k are found with the help of Maximum Likelihood Estimation (MLE).

In the test phase, the regression function is applied to the instances in the test set, predicting for each item the log odds that is should be categorized as '1'. We convert these log odds to ranks by sorting them per question. This way, we can calculate MRR for the test set.

We experiment with two different cost functions with logistic regression:

1. MLE default regression. We build a logistic regression function using all 37 features from our set as input, without interactions.
2. MLE stepwise regression. We use a recursive wrapper that in each step adds the significant feature that gives the highest improvement in terms of MRR for the training set. Then it builds an MLE regression function using the newly added feature and the features kept from previous rounds. The wrapper stops adding features once no improvement is gained anymore or no significant features are left according to the MLE regression module.

5.3.3 SVM

In order to assess the performance of discriminative models of classification (as opposed to regression-based models) to our data, we investigate the use of support vector machines (SVMs) [4].

We use version 6 of SVM^{light}[14] for training and testing support vector machines. In the testing phase, SVM^{light} assigns a score to each of the data instances. We convert these scores to ranks by sorting them per question. We first experiment with linear and polynomial kernels.

We use SVM^{light} in two different cost functions:

1. Classification. SVMs attempt to derive the hyperplane that optimally separates data in different classes (with a margin as large as possible). This hyperplane is described by a linear function in a high dimensional space. The optimization problem for SVMs consists of finding a weight vector $\vec{w}$ and a constant b, such that $\frac{1}{2} \mid\mid \vec{w} \mid\mid^2$ is minimized w.r.t. c_i $((\vec{w} \cdot \vec{x}_i - b) \geq 1$, $(1 \leq i \leq n)$, in which $\vec{x}_i$ a data vector, and $c \in \{1, -1\}$ the class of $\vec{x}_i$).
2. Pairwise preference learning (Ranking SVM [9])[15].

6. RESULTS

The results that we obtained using the different machine learning techniques and optimalization functions are in Table 2. In the case of SVM, we only show the results obtained with the linear kernel, since a polynomial kernel did not improve the results. For all settings, success@150 is 78.5%. This score does not change by re-ranking the results. For significance testing, we used the Wilcoxon Signed-Rank test on paired reciprocal ranks.

Table 2 shows that the best results are obtained with logistic regression (both MLE default as stepwise MLE optimizing MRR) on data that were vertically normalized per question (rows 5 and 7 respectively). Although vertical normalization per question seems to give better results than horizontal vector normalization for all settings, the difference is only significant ($p = 0.024$) for MLE default logistic regression (rows 4 and 5 compared).

If we compare the results for the different settings on the horizontally normalized data, we see that stepwise logistic regression (row 6) and the genetic algorithm optimizing

[12]In these experiments, we set generation size to 500 and the number of generations to 50.

[13]See http://cran.r-project.org/web/packages/Design/index.html/

[14]See http://svmlight.joachims.org/

[15]We are aware of the fact that Ranking SVM expects multilevel relevance as opposed to our binary labeling, but it still is interesting to see what can be done with pairwise preference ranking.

Table 2: Results for all learning settings in terms of MRR and Success@10. Success@150 is equal for all settings: 78.5%. An asterisk (*) indicates a statistically significant improvement ($P < 0.01$ according to the Wilcoxon Signed-Rank test) over the baseline. For each evaluation measure, the highest score is printed in bold face.

	Feature set	learning algorithm and cost function	MRR	success@10
1	TF-IDF (baseline)	-	0.249	45.21%
2	37 feats, horizontal norm	genetic, scores to ranks, optimizing MRR	0.309*	53.48%
3	37 feats, horizontal norm	genetic, pairwise preference learning, optimizing Tau	0.301*	54.01%
4	37 feats, horizontal norm	logistic regression, MLE default	0.273	51.43%
5	37 feats, vertical norm	logistic regression, MLE default	**0.341***	**58.82%**
6	37 feats, horizontal norm	logistic regression, MLE stepwise, optimizing MRR	0.301*	56.14%
7	37 feats, vertical norm	logistic regression, MLE stepwise, optimizing MRR	0.328*	56.69%
8	37 feats, horizontal norm	SVM, linear kernel, pairwise preference learning	0.048	6.95%
9	37 feats, horizontal norm	SVM, linear kernel, classification	0.053	11.62%

MRR (row 2) give similar results: MRR is around 0.305. Default MLE logistic regression (row 5) on the horizontally normalized data does not give significant improvement over the baseline: MRR is 0.273.

SVM (both classification and ranking with either a linear or a polynomial kernel) performs much worse than that and even lower than baseline: MRR for SVM is around 0.05.

7. DISCUSSION

After we have excluded the possibility of bugs in our experimental set-up for SVM, we follow up with a series of experiments to find out where the bad results with SVM come from and what kind of kernels and hyperparameters are needed for improving them. This is discussed in Section 7.1 below. In Section 7.2 we look at the best-scoring machine learning techniques and cost functions, and present the features they deemed most important.

7.1 More experiments with SVM

We consider three possible causes for the poor results we obtained with SVM: the presence of complex relations between the features in our set, the complexity of the features themselves, and the class imbalance in our data collection (much more negative than positive instances).

7.1.1 Complex feature relations

In Section 4, we pointed out that we use complex (structural and semantic) features. Some of our features are correlated and others even depend on each other. Therefore, we now experiment with a highly simplified version of our feature set in order to find out whether the complex feature relations cause the poor results. To this end, we removed all features except TF-IDF. If we use logistic regression to build a model for the training set using TF-IDF only and we apply this model to the test set, the probabilities assigned to the instances lead to baseline ranking (MRR around 0.24). However, if we try this with SVM, we still get very poor results (MRR around 0.07, depending on the hyperparameter setting we choose). This means that the low scores that we obtain with SVM are not due to complex relations between the features in our set.

7.1.2 Complex features

In order to find out whether the poor results are caused by the complexity of the features themselves[16], we experiment with a set of simple surface features. For every question-answer pair, we create a bag of WordNet expansions. The bag contains all WordNet synonyms, hypernyms, hyponyms, senses, and antonyms for the nouns and verbs in the question and in the candidate answer. For every WordNet expansion word in the bag, we counted its frequency in the bag and L1-normalize it. These L1-normalized frequencies constitute the feature vector for the question-answer pair. It is a probability distribution summing to 1.

L1-normalized data is most naturally learned by multinomial kernels, also known as information diffusion kernels [11]. These are kernels that deploy geodesic distance measures on L1-normalized data. Lafferty and Lebanon [11] argue that geodesic distance is often a better approximation of the information geometry of L1-normalized documents than plain Euclidean distance. Previous work [14] demonstrates that multinomial kernels are able to produce state of the art results for sentiment polarity classification tasks.

In SVM^{light}, we implement a simple, hyperparameter-free multinomial kernel, i.e. the shifted negative geodesic kernel K_{NGD} [19]:

$$K_{NGD}(\vec{x},\vec{y}) = -2\arccos\left(\sum_{i=1}^{n}\sqrt{x_i y_i}\right) + \pi \qquad (6)$$

in which $\vec{x}$ is a support vector from the training data, $\vec{y}$ is a feature vector from the test data, and $i_1, .., i_n$ is the set of features occurring in both $\vec{x}$ and $\vec{y}$.

With this kernel, we again obtain a result of $MRR = 0.07$ for the surface WordNet features. This indicates that the complexity of the features does not cause the low scores.

7.1.3 Class imbalance

In Section 4, we pointed out the problem of class imbalance in our data. As a first option for solving this problem, we vary the cost-factor in SVM^{light} by which training errors on positive examples outweight errors on negative examples[17].

[16] TF-IDF itself is a complex measure that combines a number of counts in one function.

[17] See the documentation of SVM^{light} for the implementation of the cost-factor.

We experiment with a cost-factor of 10 and 100. This does however not improve our results in terms of MRR, which were still around 0.05.

Next we opt for another method: a sampling based approach akin to bootstrap aggregating or bagging [1]. We again use the simple training data that we described in Section 7.1.2. We sample the training data for a number of n times, drawing with replacement a number of exactly k items from the training data, among which, for every sample, are all relevant instances (labeled '1'). For every such sample, a separate classifier is trained and applied to the test data, after which the decision values of all classifiers are averaged to produce the final result. The optimal values of n and k are determined through grid search. For $n = 5$ samples of $k = 400$ items we obtain a result of MRR=0.147. This result, while still well below baseline, at least demonstrates that the original SVM results can be significantly improved by tackling the class imbalance.

7.1.4 Support Vector Regression

We also experiment with support vector regression[18] to find out if support vectors can be used for training regression functions for our 37-features data. We use SVMlib[19] for these experiments. We normalize the feature values vertically per question since we found in Section 6 that this gives better results than horizontal vector normalization. We choose the following parameter values: $c = 0.00195, \gamma = 0.000122, \nu = 0.1$ and we used an RBF kernel[20]. With these settings, we obtain an MRR score of 0.338 with success@10 57.75%, which is similar to the results obtained with logistic regression.

7.1.5 Future suggestions

Since we learnt that the poor results from SVM can at least partly be explained by the extreme class imbalance, we plan to apply the bagging method to our original set of 37 features in the near future.

We have one other suggestion for future experiments with SVM: The current setup treats the answer ranking problem essentially as a binary classification problem: questions are paired with answers, and the possible outcomes of this pairing are '0' (irrelevant) and '1' (relevant). From this binary classification problem, we tried to deduce a ranking by treating the binary ground truth as the discretization of a continuous decision function. This effectively may not be the best option, hampering, for one thing, the use of ranking classifiers such as Ranking SVM, which presuppose ranked ground truth. It would seem that this forced way of learning a ranking from binary data only aggravates the problem of class imbalance in our data.

7.2 Important features

[18]See http://svms.org/regression/

[19]See http://cs.haifa.ac.il/YOSI/PCOMP/

[20]We tested linear and RBF kernels with various parameter settings across the possible range of settings for the first training fold. This way, we obtained oracle parameters for this fold. Then, we applied the same parameter settings to the other training folds. We unfortunately did not have time for optimizing parameter settings for all training folds.

Both the genetic algorithm and the stepwise regression approach give good experimental results. This makes it interesting to see which features made the improvement. In order to find out which features are the most important for ranking the answers using the genetic algorithm, we selected the features that were assigned an average weight larger than 7 with a standard deviation smaller than 2 over the five folds (see Table 3). We also had a look into the features that were selected as significant features in at least two of the five folds in the stepwise regression approach (see Table 4).

Table 3: The features that were assigned an average weight > 7 with a standard deviation < 2 over the five folds by the genetic algorithm. Behind each feature in Table 3 is the average weight that was assigned to the feature over the five training folds.

Feature	Average weight
nonfocus overlap	10
TF-IDF	9.8
cue words	9.8
verb synonym overlap	9.6
doctitle focus synonym overlap	7.4

Table 4: The features that were selected as significant features in at least two of the five folds in the stepwise regression approach.

Feature	# folds
TF-IDF	5
doctitle focus synonym overlap	5
doctitle focus overlap	4
WordNet relatedness	3
head overlap	3
passage position	2
doctitle synonym overlap	2

There are a few differences between Table 3 and 4, showing that similar results can be obtained with different subsets of our features. This is partly due to feature redundancy: the same information is sometimes described by two different features. E.g. the question's main verb is always the head of a verb phrase. Therefore, the feature 'head overlap' represents partly the same information as the feature 'verb overlap'.

We see in Table 3 that the presence of cue words can give useful information in re-ranking answer paragraphs[21]. In fact, incorporating the presence of cue words is a step towards recognizing that a passage is potentially an answer to a *why*-question. As argued in Section 1, identifying a passage as a potential answer is the important issue in *why*-QA, since answers cannot be recognized by simple semantic-syntactic units such as named entities as is the case for factoid QA.

In both feature selections, we see the importance of question focus and document title. The importance of question focus for *why*-QA is especially interesting because it is a question feature that is specific to *why*-questions and does not similarly apply to factoids or other question types. Moreover, the overlap between the question focus and the document

[21]In the stepwise regression approach, this feature was only selected in one of the five folds.

title shows that Wikipedia as an answer source can provide QA systems with more information than a collection of plain texts without document structure does. In Table 3, we see that the overlap between the non-focus part of the question and the passage is also important. We can clarify this with a simple example: the question "Why are flamingos pink?" has *flamingo* as focus and *pink* as non-focus part. We can find the answer to this question in the Wikipedia document with title *flamingos*, in the passage that describes their *pink* color. In general, in cases where the question focus leads to the document title, the non-focus part often leads to the answer passage within this document.

8. CONCLUSION

We can draw two important conclusions from the current paper.

First, a learning to rank approach using either a regression technique or a genetic algorithm that optimizes for MRR leads to a significant improvement over the TF-IDF baseline. We reach an MRR of 0.341 with a success@10 score of 58.82%. Although this improvement is significant, the system is still limited to answering 59% of the *why*-questions in the top 10, while almost 80% of questions have a relevant answer somewhere in the top 150. In the near future, we plan to find out which answers are retrieved but not ranked in the top 10 and why. We also plan to investigate the 20% of the questions in our set are not retrieved by our QA system at all.

Second, we found that the results obtained with SVM are very poor compared to the results obtained with logistic regression and genetic algorithms. In future work, it would be interesting to experiment with (1) bagging techniques applied to our set of 37 features, as potential solution for the data imbalance; and (2) a ranked ground truth (multi-level instead of binary labeling), so that ranking classifiers such as Ranking SVM can be better applied to our data.

9. ACKNOWLEDGEMENTS

We would like to thank Hans van Halteren for his experiments with support vector regression. Also, we thank the anonymous reviewers for their valuable comments.

10. REFERENCES

[1] L. Breiman. Bagging predictors. In *Machine Learning*, pages 123–140, 1996.

[2] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of ICML 2005*, volume 22, page 89, 2005.

[3] B. Carterette and D. Petkova. Learning a ranking from pairwise preferences. In *Proceedings of SIGIR 2006*, pages 629–630. ACM New York, NY, USA, 2006.

[4] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000.

[5] L. Denoyer and P. Gallinari. The Wikipedia XML corpus. *ACM SIGIR Forum*, 40(1):64–69, 2006.

[6] W. Fan, E. Fox, P. Pathak, and H. Wu. The Effects of Fitness Functions on Genetic Programming-Based Ranking Discovery for Web Search. *Journal of the American Society for Information Science and Technology*, 55(7):628–636, 2004.

[7] J. Furnkranz and E. Hullermeier. Pairwise Preference Learning and Ranking. *Lecture Notes in Computer Science*, pages 145–156, 2003.

[8] E. Hovy, U. Hermjakob, and D. Ravichandran. A Question/Answer Typology with Surface Text Patterns. In *Proceedings of HLT 2002*, San Diego, CA, 2002.

[9] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of ACM SIGKDD 2002*, pages 133–142. ACM, 2002.

[10] M. Khalid and S. Verberne. Passage Retrieval for Question Answering using Sliding Windows. In *Proceedings of COLING 2008, Workshop IR4QA*, 2008.

[11] J. Lafferty and G. Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning*, 6:129–163, 2005.

[12] T. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, 2007.

[13] T. Pedersen, S. Patwardhan, and J. Michelizzi. WordNet:: Similarity-Measuring the Relatedness of Concepts. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1024–1025, 2004.

[14] S. Raaijmakers. Sentiment classification with interpolated information diffusion kernels. In *Proceedings of the First International Workshop on Data Mining and Audience Intelligence for Advertising (ADKDD'07)*, 2007.

[15] J. Tiedemann. A Comparison of Genetic Algorithms for Optimizing Linguistically Informed IR in Question Answering. *LECTURE NOTES IN COMPUTER SCIENCE*, 4733:398, 2007.

[16] A. Trotman. An Artificial Intelligence Approach To Information Retrieval. *Proceedings of the SIGIR 2004 Doctorial Consortium*, page 603, 2004.

[17] S. Verberne. Paragraph retrieval for why-question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 922–922. ACM Press New York, NY, USA, 2007.

[18] S. Verberne, L. Boves, N. Oostdijk, and P. Coppen. Using Syntactic Information for Improving Why-Question Answering. In *Proceedings of COLING 2008*, 2008.

[19] D. Zhang, X. Chen, and W. S. Lee. Text classification with kernels on the multinomial manifold. In *Proceedings SIGIR'05*, pages 266–273, 2005.

Non-Linear Query Reformulation Behavior

Nikolai Buzikashvili
Institute of System Analysis, Russian Academy of Science
9, prospect 60-Let Oktyabrya, Moscow, 117312, Russia
buzik@cs.isa.ru
+7 (499) 135-5357

ABSTRACT
In contrast to a common query log analysis approach considering a query as a possible modification only of its direct predecessor, we also consider non-linear dependencies between queries: branching, merging, re-merging (merging of earlier branched chains) and non-linear execution of linear chains. To detect short-term dependencies between user queries we use a method based on layered query similarity; dependency is considered as a time-directed similarity. The method is applied to query logs of the major search engine. The findings are: about 10% of sessions containing 3 and more distinct queries contain combinations which can be interpreted as branchings; number of branches are a little bigger than two; one of the branches consists of only one query; branching may be realized as both narrowing and broadening. About 5% of time sessions containing 3+ distinct queries contain mergings, and the merging search is usually executed in the re-merging manner. In addition to structuring of a query modification process, such properties as narrowing and broadening in each kind of structural dependencies are studied.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: *Search process*.

General Terms
Algorithms, Experimentation, Human Factors, Theory.

Keywords
Query modification process, non-linear query dependencies.

1. INTRODUCTION
The paper considers a query modification process in terms of dependencies between queries submitted by the same user and is centered on non-linear dependencies. Early conceptual works [1, 2; 5; 12] on algorithmic-like description of information searching behavior do not exclude a possibility of non-linear search: a search process may contain several branches [2] and a complex search task may be decomposed into different chains [12], which merge in the final step. However, all following empirical studies in the field of user searching behavior (e.g. [10, 11]) use only linear search framework in which a query can be described as dependent only on its direct predecessor. One of the reasons is an absence of a formal language describing non-linear dependencies and tools automatically detecting them in query logs. While [10, 11] consider user actions in different search contexts and different search environments, the works coincide in *a framework*: a current query is considered as a possible modification of only its *direct* predecessor $Q_T=f(Q_{T-1})$. Dependencies on indirect predecessors are neglected.

Works [7, 9] using big query logs (and centered on improvement of the search results by implicit feedback rather on user searching behavior as such) also use a *linear* model of query modification. On the contrary, we follow to the opposite conceptual framework proposed in [4]:

(1) a query may depend on non-direct predecessor: $Q_T=f(Q_{S<T})$ instead of $Q_T=f(Q_{S=T-1})$. As a result, several queries may depend on the same query (a *branching* search: queries $Q_{T1}=f(Q_S)$, … $Q_{T2}=f(Q_S)$ depend on the same Q_S).

(2) a query may be a "combination" of *a pair* of earlier submitted queries $Q_T=f(Q_{S1<T}, Q_{S2<T})$ (a *merging* search) rather than a modification of a single query.

Interpretations of the same series of user queries may significantly vary depending on the query dependencies taken into account. Figure 1 shows differences between linear and non-linear interpretations of two time series of user queries.

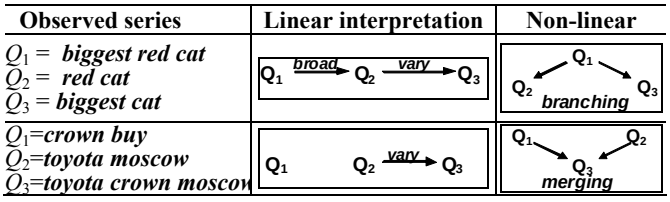


Figure 1. Observed user queries and their interpretations.

However, are the non-linearity assumptions realistic? To what extent are they (a) psychologically grounded and (b) technically supported by search engine interfaces? A branching search is not a surprising manner for any living being. A human searching on the Web comes up against a common situation provoking a branching search: when an initial query perfectly expresses an information need but retrieves unsatisfactory results, a user modifies the query; and if the results of the modified query are also unsatisfactory, a user refines the *initial* query rather than the current query.

The 9th Dutch-Belgian Information Retrieval Workshop (DIR 2009), February 2-3, 2009, Enschede, The Netherlands.

A merging search is poorly supported by a human memory and is not supported by search environments. However, a brief interviewing of common searchers shows that both manners are consciously applied when a "direct search" gives unsatisfactory results. Besides, we can expect of a lot of "unconscious" branching modifications of the initial queries perfectly presenting information needs but giving bad search results.

We suppose that a non-linear model of query modifications (cf. a broad conceptual model [6]) more adequately describes a real searching behavior that is especially important because a non-linear search is only partly supported by search engines. If a non-linear model is adequate:

(a) then a direct support of non-linear search may be incorporated into search tools.

(b) Next, to detect identical subparts in the serieses of transactions the linear model is used and only sub*sequences* are considered. However, there is no difference between different orderings of the same merging parts and no difference seems to be between sequentially executed linear chains (*chain1*, *chain2*) and *chain1* followed by [the last] branch *chain2* in branching $chain1 < \begin{matrix} chains \\ chain2 \end{matrix}$.

Thus, to detect common parts in time serieses of user's queries we should take into account that a sequence (*chain1*, chains, *chain2*) presenting branching indeed contains subseries (*chain1*, *chain2*).

2. CONCEPTUAL MODEL

All dependencies are detected between queries submitted during a short-time period (a *time session*, i.e. a series of user transactions with a search engine cut from previous and successive sessions by the time gap bigger than an assigned *cutoff*). Let dependencies between queries be presented by a graph of a *logical structure* of a search [4], in which each query be presented by a unique node regardless of a number of occurrences of the query in a session. The connected components of this graph correspond to "task sessions". The same logical structure may present different real search processes. Different tasks or different branches of the same task may alternate in execution and even linear tasks may be executed non-linearly: one task is broken by another task and restored again (*occasional non-linearity*).

Let distinct queries submitted by the same user during a time session be presented by nodes and dependencies (the strongest dependencies according to a certain dependency measure, which will be set in a form of 2-step decision rule) between queries be presented by arcs. We will refer to this directed acyclic graph as a Query Modification Graph. According to (2) a maximum indegree is equal to 2; according to (1) outdegrees are not limited. If there are two paths from the node *R* to the node *M* crossing only in *R* and *M* we speak about re-merging (merging of earlier branched chains) and call *H* a root of re-merging.

Figure 2 shows synthetic examples of basic (branching, merging) and derivative non-linearities (re-merging and occasional non-linearity of execution) in a search process and its logical structure.

An applicable non-linear model (1)–(2) presents a general framework to describe a query modification process. There are a lot of technical variants of the model implementation. Some of them are more expensive (e.g. "semantic"-based), some are less ("lexical" term-based). Here, we use several rules of structuring query modification process based on different lexical dependency measures. Different kinds of dependency measures are sound and the best way to detect and to analyze a non-linear search is to use a family of decision rules, each of which may operate with different measures (e.g., combine them) since different rules refining different aspects of a search behavior are equally feasible.

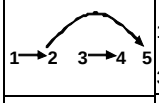
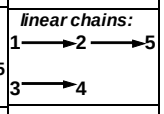
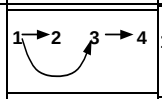
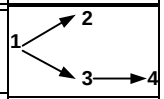
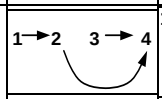
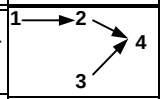
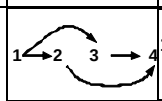
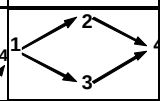
Type of non-linearity	Sequence of execution	Dependencies in execution	Dependencies in logical structure
Occasional nonsequential **execution** of linearly-dependent queries	*kitten food* **(1)** *cat food* **(2)** *DIR'*09 **(3)** *DIR 2009* **(4)** *hills cat diet* **(5)**		
Branching (one query determines several ones)	*cat food* **(1)** *kitten food* **(2)** *home for cats* **(3)** *home for dogs***(4)**		
Merging (a query depends on a pair)	*kitten food* **(1)** *cat food* **(2)** *hills* **(3)** *hills cat diet* **(4)**		
Re-merging (of earlier branched chains)	*cat food* **(1)** *kitten food* **(2)** *DIR cats* **(3)** *cat kitten DIR* **(4)**		

Figure 2. Non-linearities of logical structure and realization.

3. DATASET

A week sample (March, 2005) of the query log of the Russian search engine *Yandex* is used[1]. The dataset was pre-processed to exclude users who are robots rather than humans. To do it a client discriminator threshold equal to 7 unique queries per 1-hour sliding window was applied. 30-min intersession cutoff was used to segment user transactions into time sessions. The preprocessed dataset (Table 1) contains 117,097 users executed 644,901 transactions. Time sessions containing query language operators (in particular quotations) were excluded to simplify processing.

Table 1. Preprocessed dataset

time sessions	225,451
time sessions containing one query	138,757(61.55%)
time sessions containing 2+ distinct queries	86,694 (38.45%)
time sessions containing 3+ distinct queries	38,571 (17.11%)
non-first queries in time sessions	156,101
non-first-two(= non-last-two) queries in t.sess-s	66,926

[1] The *Yandex* logs are the best representative of Russian-language queries. While the fraction of queries submitted to the *Yandex* from the ex-USSR monotonically decreases the *Yandex* remains the most popular search engine. For example, according to http://www.liveinternet.ru/stat/ru/searches.html (January 2009), 46% of the Russian domain (.ru) sites are visited from the *Yandex*, 34% — from *Google* (cf. 1.5% for *MS search*).

4. METHOD

4.1 Method in Brief

Each non-first query submitted during a time session is considered as possible dependent query, which may be determined either by a single query or a pair of queries submitted earlier during the session. Dependency on a pair is fixed only if this dependency is bigger than dependency on any single query. Different measures are used to select 1- and 2-dependencies. When a dependent query Q is attributed accordingly to used dependency measures, decision rules are applied to select a "main" determinant of Q among all of its determinants. A decision rule may use one or several 1- and 2-dependency measures. While a decision rule may combine different measures and use simultaneously 1- and 2-measures, we apply 2-step decision rules: 1) in the first step the main 1-determinant Q^{Det} is chosen as the most recent among queries which have the most influence on Q according some measure; 2) if the chosen 1-determinant Q^{Det} belongs to a pair (Q^{Det}, Q^{Det*}) mostly determining Q according to the measure used in this step (this measure may differ from the measure used in the first step to choose 1-determinant), the pair (Q^{Det}, Q^{Det*}) is selected as a final determinant; otherwise Q is determined by a 1-determinant Q^{Det}.

4.2 Layered Query Image

A *query image* is an unordered set of query terms. To determine inter-query dependencies we use layered query image and layered processing [3]. According to this approach, terms of a query are divided into several layers which contain different role classes (Table 2) containing different parts of speech (POS). Unambiguous POS tagging is not a problem in inflecting languages (e.g. Russian); non-dictionary words are effectively attributed heuristically [3]. Terms are not ordered inside layers.

Table 2. Role classes of parts of speech

Role Class	Parts of speech included into Class
Subjects / Objects	nouns, names, acronyms, + unknowns which may be nouns, names, acronyms
Features	adjectives, participles, [+ unknowns which may be adjectives, participles] numerals and numbers
Actions	verbs, adverbs, adverbial participles [+unknowns which may be these POSs]
Others	all other words (preposition, article, particle, conjunction, interjection, etc)

The role of the core layer $L1$ is special, an intersection of core layers determine a *fact* of query similarity: if the *core layers* $L1(Q)$ and $L1(Q')$ of queries Q and Q' do not intersect, Q and Q' are considered as dissimilar and intersections of outer layers are ignored. To choose query(ies) to which a query Q is the most similar, layers are processed on-by-one: if according to used similarity metric a query Q is more similar to Q_1 than to Q_2 *on sets of query terms belonging to the layer* Li then Q is considered as more similar to Q_1. A set of considered queries $\{Q_i\}$ is reduced in each step: if Q is the most similar to just one query, a in-layer similarity detection is terminated, otherwise if Q is maximally similar to several queries $\{Q_1,\dots Q_k\}$, similarities of Q and *these* queries are compared in the next layer, and so on.

Example. Let the *overlap* metric be applied to 2-layer representation $L1$={*Subjects*), $L2$= (*Features*+*Actions*+*Others*). We should detect queries which are the most similar to Q=<*big cat walks*> among Q_1=<*big brother walks*>, Q_2=<*cat*> and Q_3=<*cat jumps in big garden*>. $L1(Q)\cap L1(Q_1)=\emptyset$, that is Q and Q_1 are dissimilar. $|L1(Q)\cap L1(Q_2)| = |L1(Q)\cap L1(Q_3)| =1$. Since two queries are maximally similar to Q, we consider the metric in the next layer of these queries: $L2(Q)\cap L2(Q_2)=\emptyset$, $L2(Q)\cap L2(Q_3)=\{big\}$. Thus, Q_3 is the most similar to Q.

[3] shows that the results of layered query processing slightly smaller than the results of non-layered processing (about 5-10% of "similarities" detected in non-layered processing are not detected in more strong layered procedure of query-to-query similarity detection). To detect dependency we use the 3-layered query representation: $L1$={*Subjects*}, $L2$={*Features* and *Actions*}, $L3$={*Others*}.

Distribution of POS-based classes in queries obviously differs from term distributions in documents but it doesn't contain only *Subjects* and *Features*: 74.19% of terms belong to the *Subjects* class, 14.39% to the *Features* class, 3.55% to the *Actions* and 7.87% to the *Others* class among 978,930 terms in 357,141 distinct queries submitted to the *Yandex*. Other query datasets show similar shift, e.g. distribution of "successful terms" in queries in Dutch [8]: 58% are nouns (the main part of *Subjects*), 29% are verbs (the main part of *Actions*) and 13% are adjectives (the main part of *Features*)

While several terms of a query may be forms of the same term they are not considered as duplicates and *each* of the terms is included into image. For example, image of the query <*cat, cat and cats*> is (*cat*, *cat*, *cats*).[2]

Let a core of a query intersect with cores of several queries submitted earlier and considered as determining. We should choose the determining query (a pair of queries in the case of merging) mostly influencing it according to a dependency measure used (or rank determinants accordingly to their influence on the dependent query).

4.3 Query Dependency Measures

A layer is unordered set of terms. Such a representation simplifies processing. At the same time, since we consider only queries submitted during a limited period it is extremely unlikely that two queries presented by the same unordered set of terms express different information needs.

We use term-based dependency measures. To detect intersections permissible word transformations are used. Semantic dependency (similarity) measures are not considered. In particular, we ignore

[2] Although the effect of this approach is small (term repetitions is very rare) it allows us to avoid a problem of identical unknown words detection significant for inflecting languages: two unknown terms $T1$ and $T2$ may be a form of the same word but third term $T3$ may be a form only of $T1$. Let similarity of Q_1=<$T1$, $T2$> and Q_2=<$T3$> be detected. If we include $T2$ into a core $L1(Q_1)$, queries Q_2 and Q_1 are recognized as dissimilar.

synonymy and aspectual relations, e.g. no dependency between <*big wild cat*> and <*jaguar*> is detected.

Dependency measures. Term-based measures of query dependency are used. There are two broad classes of term-based dependency measures:

(1) similarity metrics (intersection, symmetric difference and their normalizations in a form of Jaccard metrics) augmented by a direction from the determining query (a pair of queries) to the dependent query;

(2) originally directed dependency measures: difference, narrowing, broadening. Here, narrowing and broadening are considered in a term-based manner: narrowing is an expansion of query by new terms and broadening is an exclusion of terms (e.g. <*big cat*> → <*cat*> is broadening and <*big cat*> → <*cat and big dog*> is narrowing).

Query dependencies. Each query may depend on queries submitted previously during a considered time session. Each query may determine queries submitted later during the session. All dependency measures are applied only to images of those queries, cores of which are intersected. In the first step we check intersection of cores $L1(Q)=\{t_1,\dots t_n\}$ and $L1(Q^{\mathrm{Det}})=\{T_1,\dots T_m\}$ of a possible dependent query Q and a possible 1- or 2-determinant Q^{Det}. If $L1(Q)\cap L1(Q^{\mathrm{Det}})\neq\emptyset$, Q depends on Q^{Det}. If the only determinant Q^{Det} maximizes dependency measure in the core layer, it is chosen as a main determinant. If several determinants $\{Q^{\mathrm{Det}}\}$ maximize dependency measure, we compare dependencies on *these* determinants in the outer layer $L2$. If in turn several of them maximize dependency in $L2$, we choose the recent determinant (see next subchapter).

We use two types of dependency: on a single query and on a pair of queries considered as an entity. All measures may be used both as 1-dependency and 2-dependency measures. Besides, there is a special form of 2-dependency, an empty symmetric difference (non-strict narrowing/broadening).

1-dependency. A query is concerned as a modification of a single query submitted previously in the same time session. A single determinant is called *1-determinant*.

Any 1-dependency measure of the determining query Q_k is presented by a vector M, where $m_{j<k}$ is a score ("rank") of the influence of the determining query $Q_{i<k}$ on Q_k among all predecessors of Q_k. If Q_k does not depend on Q_i, m_j is not defined.

Example 1 (1-dependencies). Let $L1(Q1)=\{A,B\}$, $L1(Q2)=\{A\}$, $L1(Q3)=\{B\}$ and $L1(Q4)=\{B,C\}$ be cores of the queries in Figure 4. (To reduce illustrations we suppose that all queries in the example hereinafter include only core terms). Row dependency vectors for *overlap* ($V^{\cap}$), *Jaccard* (V^{Jaccard}), *symmetric difference* (V^{Δ}), *broadening* (V^{broad}) and *narrowing* (V^{narr}) measures are:

	$Q1$	$Q2$	$Q3$
$V^{\cap}(Q2)=$	<1>		
$V^{\cap}(Q3)=$	<1	.>	
$V^{\cap}(Q4)=$	<1	.	1>

	$Q1$	$Q2$	$Q3$
$V^{Jaccard}(Q2)=V^{\Delta}(Q2)=$	<1>		
$V^{Jaccard}(Q3)=V^{\Delta}(Q3)=$	<1	.>	
$V^{Jaccard}(Q4)=V^{\Delta}(Q4)=$	<2	.	1>

	$Q1$	$Q2$	$Q3$
$V^{broad}(Q2)=$	<1>		
$V^{broad}(Q3)=$	<1	.>	
$V^{broad}(Q4)=$	<.	.	.>

V^{narr} (*any query*) = <0>

2-dependency (merging). A query is considered as a merge of a pair of previously submitted queries. The determinative pair is called *2-determinant*. While a model does not limit a number of branches, it prohibits merging of more than two queries. This restriction is psychologically motivated. Merging chains are executed in parallel but even execution of two chains in parallel is a hard job.

A determining pair of queries in 2-dependencies is considered as an entity. Therefore we can define a layer of the pair as a union of terms of this queries attributed to the cognominal layers. For example, if $L1(Q_1)=$ {*cat*, *cheese*} and $L1(Q_2)=$ {*mouse*, *cheese*} then a core $L1(Q_1,Q_2)$ of the pair is {*cat*, *cheese*, *cheese, mouse*}.

2-dependency of a query Q on a pair (Q^{Det_1}, Q^{Det_2}) is considered *only if* Q depends on *each* of these queries, i.e. *both* intersections $L1(Q)\cap L1(Q^{Det\_1})$ and $L1(Q)\cap L1(Q^{Det\_2})$ are not empty. E.g., we do not consider dependency of Q_3=<*big red cat*> on a pair of Q_1=<*cat*> and Q_2=<*big red frog*> since the intersection of $L1(Q_3)=\{cat\}$ and $L1(Q_2)=\{frog\}$ is empty.

2-dependency of Q on a pair (Q^{Det1},Q^{Det2}) is fixed if $m(Q,(Q^{Det1}\cup Q^{Det2}))$ is stronger than each of cognominal 1-dependencies $m(Q,Q^{Det1})$, $m(Q,Q^{Det2})$. Furthermore, the results reported in Chapter 5 are elaborated under the strongest condition of 2-dependency detection: $m(Q,(Q^{Det1}\cup Q^{Det2}))$ should be stronger in the *core* layer(e.g. $|L1(Q)\cap L1(Q^{Det1}\cup Q^{Det2})|>[L1(Q)\cap L1(Q^{Det\,i})|$ for overlap measure). As a result, we do not detect $Q3$ as merging of $Q1$ and $Q2$ in series $Q1$=<*big cat*>, $Q2$=<*fat cat*>, $Q3$=<*big fat cat*>.

Any 2-dependency measure of the determining query Qk is presented by a triangle matrix M, where m_{ij} is a score ("rank") of the influence of determining pair (Q_i, Q_j) on Q_k among all pairs ($Q_{i<k}$, $Q_{j<i}$) according to the considered measure.

It is possible that none of the elements of the pair is a top 1-determinant (see the next example). However, is possible only for very long queries and is uncommon.

Example2 (2-dependencies). Now let's consider 2-dependency for session in Figure 4. Let $L1(Q1)=\{A,B,C\}$, $L1(Q2)=\{D,E,F\}$, $L1(Q3)=\{A,B,D,E\}$ and $L1(Q4)=\{A,B,C,D,E,F\}$ (due to unusual length of $Q4$ this is an atypical example). The matrixes describing 2-dependency of $Q4$ are:

$$M^{\cap}(Q4)=M^{Jaccard}(Q4)=M^{\Delta}(Q4)=\begin{bmatrix} & Q1 & Q2 \\ Q2 & 1 & \\ Q3 & 2 & 2 \end{bmatrix}$$

(here the top 1-determinant $Q3$ is not part of the top 2-determinant ($Q1$, $Q2$)). The image of the pair determinant ($Q1$, $Q2$) is identical to the image of the dependent query $Q4$. Matrixes of non-strict narrowing/broadening are:

$$M^{narrow}(Q4)=\begin{bmatrix} & Q1 & Q2 \\ Q2 & 2 & \\ Q3 & 1 & 1 \end{bmatrix} \qquad M^{broad}(Q4)=\begin{bmatrix} & Q1 & Q2 \\ Q2 & 1 & \\ Q3 & . & . \end{bmatrix}$$

Branching induced by 2-dependency accounting. It is worthy of note that accounting of a 2-dependency (a merge) is nothing than adding a new entering edge $Q^{Det2} \rightarrow Q$ to a dependency graph containing $Q^{Det1} \rightarrow Q$. If a graph contains an outgoing edge $Q^{Det2} \rightarrow Q^*$ then Q^{Det2} becomes a root of branching. It there is a path from Q^{Det2} to Q then a re-merge is induced. While the length of a path added by merging accounting is equal to 1 (Figure 3), the initial path from Q^{Det2} to Q should be at least 2 or longer.

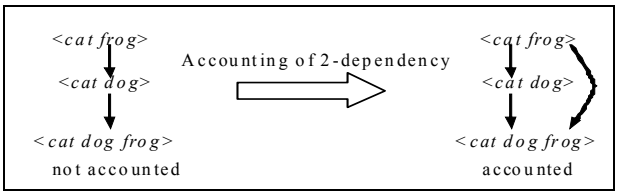


Figure 3. Branching induced by merging accounting.

4.4 Precedence of Determining Queries

Let several determinants have a maximum influence on the dependent query. Which of them should be chosen as the "major" determinant?

We consider dependency of a query on queries submitted earlier. While dependencies between *queries* should be extracted, a search engine query log contains not queries but *transactions* which correspond either to query submissions or to paging the retrieved results (Figure 4, where (Q, p) denotes a transaction with p-th page of the results retrieved by query Q).

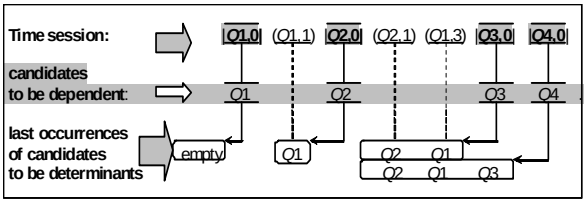


Figure 4. Constructing precedence vectors.

When a query Q is considered as *a dependent query* we consider dependencies only for the *first occurrence* of this query. When a query Q^{Det} is considered as *a determinant* of a possibly dependent query Q we take into account the *most recent* occurrence of Q^{Det} before the first occurrence of Q. How long may be a distance between occurrences of the determinant and dependent query measured in time or events between them?

A lot of queries are contextual and stored (if stored) in the limited human working memory (searchers commonly do not use external, supported by a search tools short-term query history). While we do not suppose that to "reformulate" a query a user should consciously remember the initial query, we should take into account human memory limitations. The approach used in this paper is to consider any query submitted *during a time session* as a possible determinant of a query submitted later *in this* session. Thus, dependencies only between queries submitted during the same *time session* are extracted.

Dependencies of the same queries in different sessions may be different. For example, let Q_1=<*cat*>, Q_2=<*dog*>, Q_3=<*cat and dog*> occur in 3 time sessions: <Q_1,Q_3,Q_2>, <Q_1,Q_2,Q_3>, <Q_3,Q_1,Q_2>. Q_3 is an element of a linear chain in the first session, a merge in the second and a root of branching in third.

The same query may occur several times during the session. We suppose that 1) when a query is considered as a possible depending query we should consider only *first* occurrence of the query; 2) when a query Q^{Det} is considered as a possible determinant of a query Q we should consider the recent occurrence of Q^{Det} before first occurrence of Q.

Precedence in 2-dependencies. When 1-dependency is considered there is no problem to detect the recent occurrence of a determinant (preceding the first occurrence of a dependent query) in the time-ordered series of transactions. To select the recent pair (preceding the first occurrence of a dependent query) among several determining pairs we use the following tricks:

Let (t_1, t_2) be the recent occurrences of elements of the pair, where t_j is presented either by absolute time of the recent (before a dependent query) occurrence of Q_j or by a rank of this occurrence in time-ordered scale of transactions (we use ranks.) A pair (Q_1, Q_2) is considered as more recent than (Q'_1, Q'_2) if $t_1+t_2 > t'_1+t'_2$. If several determining pairs $\{(Q^i_1,Q^i_2)\}$ maximize a sum $t^i_1+t^i_2$ we choose a pair $i|\ t^i_{1or2} = \max(\min(t^i_1,t^i_2))$, i.e. a pair the earlier element of which is the latest among earlier elements.

Example of precedence accounting. A precedence row vectors for (possibly) dependent queries $Q2$, $Q3$ and $Q4$ (calculated for their first occurrences in a time session) in Figure 4 are

$$\begin{array}{lcccc} & Q1 & Q2 & Q3 \\ \mathrm{Prec}V(Q2) = & <1> & & \\ \mathrm{Prec}V(Q3) = & <1 & 2> & \\ \mathrm{Prec}V(Q4) = & <2 & 3 & 1> \end{array}$$

and the precedence triangle matrixes (constricted on the precedence vectors) are

$$\mathrm{Prec}Matr(Q3) = \begin{bmatrix} & Q1 \\ Q2 & 1 \end{bmatrix} \qquad \mathrm{Prec}Matr(Q4) = \begin{bmatrix} & Q1 & Q2 \\ Q2 & 3 & \\ Q3 & 1 & 2 \end{bmatrix}$$

4.5 Final Dependency Extraction

Decision rules. Decision rules select a "main" determinant of each dependent query among all of its determinants. A decision rule may use one or several 1- and 2-dependency measures. A decision rule operates with 1-dependency vectors, 2-dependency matrixes and a precedence vector.

Different kinds of dependency measures are sound and the best way to detect and to analyze a non-linear search is to use a family of decision rules, which operate with different measures (in particular, combine them) since different rules refine different aspects of a search behavior and may be equally feasible.

Just one 1- or 2-determinant should be selected by the decision rule. However, 1-determinant may be not a part of 2-determinant even for the same measure (*Example* 2). To avoid a (prohibited) selection of 3 determining queries the decision rule should coordinate extraction of final 1- and 2-dependencies, which may be done in a variety of ways. Decision rules may simultaneously use different measures. While decision rules applied below are constructed as a sequential two-step procedure (1-dependency

filter is used in the first step and 2-dependency filter is applied to the results of the first step) decision rules may have another form.

Example of the final determinant selection. Let's return to Example 1 and consider a family of decision rules each of which uses one of 1-dependency measures and selects for each dependent query the most recent among queries maximally influencing it according to the measure. The dependency structures extracted by these rules are presented by:

$$R^{\cap} = R^{Jaccard} = R^{\Delta} = \begin{bmatrix} & Q1 & Q2 & Q3 \\ Q2 & 1 & & \\ Q3 & 1 & & \\ Q4 & & & 1 \end{bmatrix} \Rightarrow \quad Q1 < \begin{matrix} Q2 \\ Q3 \rightarrow Q4 \end{matrix}$$

$$R^{broad} = \begin{bmatrix} & Q1 & Q2 & Q3 \\ Q2 & 1 & & \\ Q3 & 1 & & \\ Q4 & & & \end{bmatrix} \Rightarrow \quad Q1 < \begin{matrix} Q2 \\ Q3 \end{matrix}$$

5. RESULTS

Let's consider the results yielded by two 2-step decision rules. Each of rules uses the same measure for 1- and 2-dependency determinants detection:

— ***maximal overlap (max∩)*** rule: the recent determinant *Det* (a single query Q^{Det} for 1-dependency and a pair (Q^{Det1}, Q^{Det2}) for 2-dependency) is chosen among all queries which maximize *overlap* measures *max*∩ (*Q*, *Det*);

— ***minimal symmetric difference (min*Δ*)*** rule: the recent among queries which minimize *symmetric difference* measure. In addition, the rule based on the *Jaccard* measure was used. The results of the *Jaccard* rule are practically identical to the results of of the *min*Δ rule and are not reported.

The next two subchapters present results yielded by decision rules accounting respectively only 1-dependencies (subchapter 5.1) and both 1- and 2-dependencies (subchapter 5.2). The results yielded by rules accounting only 1-dependencies (i.e. linear and branching) are subsidiary and are needed mainly to determine a contribution of 2-dependency accounting into detected branching dependencies.

When a decision procedure detects non-linear dependencies, two kinds of errors take place: false acceptance (Figure 5) and false rejection. A brief manual check of the non-linearity detection shows that *all* non-linear query reformulations correctly detected by *min*Δ rules are a subset of non-linearities detected by *max*∩ rules. That is we should commit to the results of *max*∩ rules.

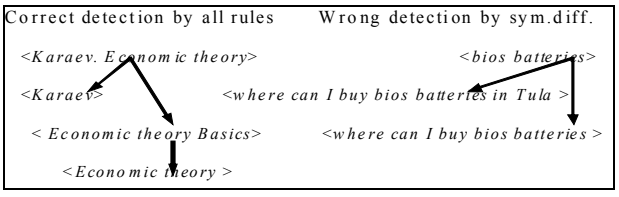


Figure 5. Real-life examples of correct and incorrect non-linearity detection.

5.1 Extraction of 1-dependencies: Branchings and Linear chains

At this step we use rules which ignore 2-dependencies.

Table 3. Characteristics of search structures extracted by 1-dependency rules *

Characteristics	*max*∩	*min*Δ
Time sessions:		
Linear time sessions: – in sessions of 2+ queries – in sessions of 3+ queries	74427 85.85% 68.20%	
Time sessions containing non-linearly executed linear chains **	7.36%	6.67%
Time sessions containing branching **	10.05%	19.70%
Queries		
Independent queries (% of non-first queries in time sessions)	68,847 (44.10%)	
Dependent on non-direct predecessor (% of non-first-two queries in time sessions)	7,717 (11.53%)	13313 (19.89%)
Roots of branching (% of non-last-2 queries = non-first-2 queries)	3991 (5.96%)	8423 (12.59%)
Childs of branching (% of non-first-2 queries)	8256 (12.34%)	18062 (26.99%)
Sons per root of branching	2.07	2.14
Length of linear chains in:		
— linearly executed linear chains	1.25	1.25
— non-linearly executed linear chains	1.49	1.34
— non-empty pre-branching linear chain (rate of empty pre-branchings)	1.30 (66.3%)	1.22 (80.3%)

* All lengths are measured in number of query modifications $Q^{(i)} \rightarrow Q^{(i+1)}$.

Structure of query modification process. Table 3 presents structural characteristics of search processes reconstructed by *max*∩ and *min*Δ rules, and the *min*Δ rule detectss much more cases of branching. We report fractions of branching sessions among time sessions containing 3 and more distinct queries (i.e. among sessions which *may* contain non-linearties) in brackets. <Несмотря на ошибки of acceptance> a branching search a frequent manner of execution of *several-query* tasks. Branching commonly has a binary form (2.1 branches per branching). One of about two branches usually contains only one query.

Table 4. Fractions (%) of 3 types of query modification in search structures extracted by 1-dependency rules *

Characteristics of	*max*∩ rule			*min*Δ rule		
	narrow	broad	other	narrow	broad	other
linearly executed linear chains	41.15	15.75	43.10	41.15	15.75	43.10
non-linearly exec. linear chains	34.48	13.76	51.76	36.15	15.33	48.51
direct pre-branching	49.0	8.0	42.9	24.2	22.5	53.4
branching root → son	14.9	29.6	55.5	50.5	11.3	38.2
branching root → union of sons	41.7	3.3	54.9	64.2	0.6	35.2

Properties of query modification in different search substructures. Table 4 presents narrowing (an image of the

determining query is a subset of the dependent query image) and broadening (an image of the dependent query is a subset of the determining query image) relations in different structures of query modification process. As Tables 4, 6 show, *max*∩ and *min*Δ rules reveal the opposite properties in non-linear query reformulation. Since is more perfect we commit to its results.

5.2 Extraction of 1- & 2-dependencies: Branching, Merging, Re-Merging, Linearities

Now we consider slight modifications of rules used to detect 1-dependencies. Two-step decision rule is used. The results of 1-dependency extraction are applied as a filter to extract 2-dependency. Namely, when the top 1-dependency determinant is detected the rule extracts the most recent and influencing determining pair (if exists), which contains the selected 1-determinant as an element. In the first step, the decision rules used in 2-step procedure are identical to the decision rules used in 1-dependency extraction. In the second step (selection of the pair) all 3 decision rules use the same *overlap* (2-dependencu) measure, i.e. the determining pair is selected among the pairs which are most influencing according to this measure and contain the top 1-determinant extracted by the 1-dependency decision rule.

Structure of query modification process. The results relating to the structure of query modification process reconstructed by the 2-step decisions rules are reported in Table 5.

Table 5. Characteristics of search structures extracted by 1- and 2-dependency rules *

Characteristics	*max*∩	*min*Δ
Time sessions:		
— linear time sessions **	67.61%	
— time sessions containing non-linearly executed linear chains **	6.27%	5.65%
— time sessions containing branching **	13.75%	22.83%
— time sessions containing merging **	9.13%	6.68%
— t.sess containing branching&merging**	5.85%	4.08%
— *time sessions containing re-merging* **	5.20%	2.88%
Queries:		
— independent queries (% of non-first queries in time sessions)	68847 (44.10%)	
— dependent on non-direct predecessor (% of non-first-two queries)	15.55%	22.60%
— roots of branching (% of non-last-2 queries = non-first-2 queries)	8.49%	14.00%
sons per root of branching	2.08	2.14
— mergings	5.71%	4.06%
Length of linear chains in:		
— linearly executed linear chains	1.34	1.25
— non-linearly executed linear chains	1.41	1.31
— non-empty pre-branching linear chain and rate of empty pre-branchings	1.26 67.4%	1.21 80.6%
— non-empty post-merging linear chain and rate of empty post-mergings	1.22 71.6%	1.17 81.0%

* All lengths are measured in number of query modifications $Q^{(i)} \rightarrow Q^{(i+1)}$.
** % of sessions containing 3+ distinct queries.

Properties of query modification. Table 6 presents narrowing (a query terms expansion), broadening and zero symmetric difference (i.e. a case of Im(*Q*)=Im(*determining Pair*) in merging) relations in query modification process.

While measures give significantly different results, we see from Table 6 that 1) the common operation in pre-branching linear modification is narrowing (query expansion), 2) narrowing is also common in branching, 3) broadening (query reduction) is common in merging and 4) narrowing is frequent in the post-merging linear modification. Narrowing is more frequent in all linear modifications.

Table 6. Fractions (%) of 4 types of query modification in search structures extracted by 1- and 2-dependency rules

Characteristics of	*max*∩ rule				*min*Δ rule			
	narr.	broad	empty Δ	other	narr.	broad	empty Δ	other
linearly exec-ed linear chains	41.17	15.76		43.07	41.17	15.76		43.07
non-linear. exec. linear chains	33.53	14.65		51.82	32.80	16.36		50.84
direct prebranch								
– non-remerging	50.3	8.5		41.2	24.4	24.2		51.4
– re-merging	58.6	6.0		35.3	40.0	12.3		47.7
direct postmerg.								
– non-remerging	37.8	20.3		41.9	25.8	3.7		70.5
– re-merging	23.0	34.8		42.2	50.1	11.8		38.1
brnch root→son								
– non-remerging	14.9	30.2		54.9	50.1	11.8		38.1
– re-merging	20.7	3.3		75.9	25.8	3.7		70.5
branch.root → union of sons								
– non-remerging	42.3	3.3		54.4	63.7	0.6		35.7
– re-merging	36.7	0.1		63.2	40.4	0.1		59.5
mergings:								
–nonREmerging	3.4	53.9	13.0	29.7	3.5	52.3	13.0	31.3
– re-merging	5.8	48.8	15.1	30.3	5.7	47.4	15.1	31.9

A *merging search*: it is hardly probable that a merging results from the conscious task disaggregation (cf. [12]). The most mergings are *re-mergings*, which typically have the very special form of a merging a branch and the *initial* query. It is likely that the initial query perfectly presents the user need but the retrieved results are bad. A user modifies the query and when the results become good he "partly returns" to the initial query. A user does not remember the initial query but he perfectly remembers the need.

Table 7. Shortest path is executed earlier than longest path in:

	max∩ rule	*min*Δ rule
branchings (1-dependency)	63.4%	67.0%
branchings	63.7%	67.1%
non-remerging mergings	40.6%	38.8%
remergings	32.1%	37.7%

Precedence of different-length paths in non-linearities. The results presented in Table 7 correspond to expectations about a probable searching behavior. The later branches in non-remerging branching correspond to the "last hope" attempts and have more chances to be longer. On the contrary, the final of two merging paths have more chances to be shorter since (1) a searcher should

not forget the current pre-merging state in the first path and (2) a goal of the final path is to "till" results of the first.

6. CONCLUSIONS

A model presented in the paper allows for query dependencies on non-direct predecessors. As a result, a modification process may be branching. Dependency on a pair of queries considered as an entity is allowed too. As a result a process may be merging and re-merging (as a merging of branched chains). It is shown that non-linear dependencies between user's queries are detected as really frequent whatever the technique used.

The results are:

(1) a branching search is a frequent manner of an execution of several-query tasks (about 10% of such time sessions contain branchings);

(2) a number of branches is a little bigger than two;

(3) one of the branches consists of only one query;

(4) not only narrowing but also broadening branching is detected.

(5) the first of [usually two] branches is usually shorter than second.

(6) the first of [perfectly two] merging paths in s usually longer than second..

While the branched search is partly supported by search environments, the merging (in particular, re-merging) search is not supported at all. However,

(7) about 6-7% of time sessions containing 3+ distinct queries contain mergings, and

(8) the merging search is frequently (3-4% of time sessions containing 3+ distinct queries) executed in the re-merging manner (merging of earlier branched chains), particularly as combining the initial query (i.e. the root of branching) perfectly presenting the information need (but retrieving unsatisfactory results) and the current query modification.

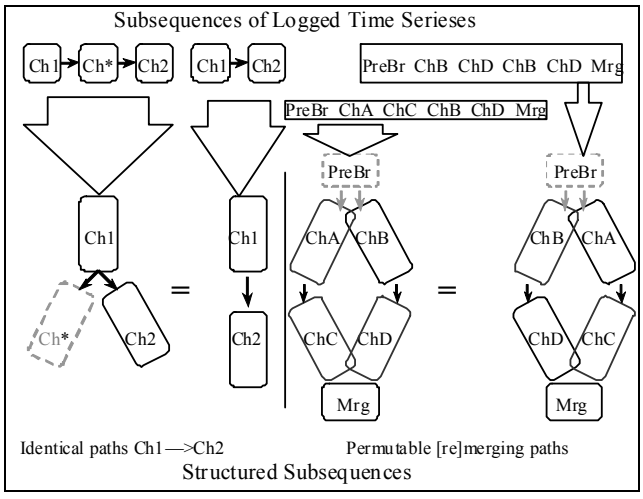


Figure 6. Examples of identical substructures.

Further work. An extraction of common parts in query logs is based on linear model and uses only methods of sequential patterns detection. However, non-linear query reformulation should be taken into account: an order of merging paths plays no part in non-remerging mergings and it is possible that paths in re-merging modification can be considered as permutable too. Whilst branches in non-merging branching are not permutable, sequentially executed chains of query modification (*Ch1*, *Ch2*) and *Ch1* followed by the final branch *Ch2* in *any* branching may be considered as identical. Figure 6 shows examples of query modification which are dissimilar within linear framework and are identical within non-linear model. The obvious next step is an modification of non-expensive sequential techniques of such identities detection (rather than an usage of general method of identical subgraphs detection).

REFERENCES

[1] Bates, M. 1979. Information search tactics. Journal of American Soc. for Inf. Sci., 30, 4 (1979), 205-214.

[2] Bates, M. 1979. Idea tactics. Journal of American Soc. for Inf. Sci., 30, 5 (1979), 280-289.

[3] Buzikashvili, N. 2008. Query-to-Query Similarity: Layered Representation and Processing. LWA, FGIR-2008 (Wurzburg, Germany, Oct. 2008), Uni.Wurzburg, 99-102.

[4] Buzikashvili, N. 2008. Structure of query modification process: branchings. ADMA'08 (Chengdu, China, Oct. 2008). LNAI, vol. 6139, Springer, 717-724.

[5] Fidel, R. 1985. Moves in online searching. Online Review, 9, 1, (1985), 61-74.

[6] Foster, A. 2004. A non-linear model of information seeking behaviour. Journal of American Soc. for Inf. Sci. and Tech., 55, 3 (2004), 228-237.

[7] Jones, R., Rey, B., Madani, O., and Greiner, W. 2006. Generating query substitutions. In Proceedings of the WWW 2006 (Edinburgh, Scotland, May 2006), ACM Press, 387-396

[8] Kraaij, W. and Pohlmann, R. 1996. Viewing stemming as recall enhancement. In : *Proceedings of the 30th annual international ACM SIGIR*, pages 40-48, Zurich, Switzerland, August 1996, ACM Press, 1996.

[9] Radlinski, F. and Joachims, T. 2005. Query Chains: Learning to rank from implicit feedback, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD'05), ACM Press, NY, 2005

[10] Rieh, S.Yu. and Xie, H.I. 2006. Analysis of multiple query reformulations on the web: The interactive information retrieval context. Inf. Proc. & Manag., 42 (2006), 751–768.

[11] Vakkari, P. 2000. eCognition and changes of search terms and tactics during task performance. In: Proceedings of RIAO'2000 (Paris, France, April, 2000), 894–907.

[12] Wildemuth, B., Jacob, E., Fullington, A., de Blieck, R., and Friedman, Ch. 1991. A detailed analysis of end-user search behaviors. In: 54th ASIS Annual Meeting, ACM Press, NY, 302-312.

Language Independent Content Extraction from Web Pages

Javier Arias Moreno
Department of computer science
Celestijnenlaan 200A
3001 Leuven, Belgium
Arias@lsi.upc.edu

Koen Deschacht
Department of computer science
Celestijnenlaan 200A
3001 Leuven, Belgium
Koen.Deschacht@cs.kuleuven.be

Marie-Francine Moens
Department of computer science
Celestijnenlaan 200A
3001 Leuven, Belgium
Sien.Moens@cs.kuleuven.be

ABSTRACT

In this paper we present a simple, robust, accurate and language-independent solution for extracting the main content of an HTML-formatted Web page and for removing additional content such as navigation menus, functional and design elements, and commercial advertisements. This method creates a text density graph of a given Web page and then selects the region of the Web page with the highest density. The results are comparable or better than state-of-the-art methods that are computationally more complex, when evaluated on a standard dataset. Accurate and efficient content extraction from Web pages is largely needed when searching or mining Web content.

General Terms

Web page cleaning

1. INTRODUCTION

When building a system for searching or mining Web content, a first task is extracting the main content and removing extraneous data such as navigation menus, functional and design elements, and commercial advertisements. Also when showing Web pages on small screens (e.g., of mobile phones) or sending text to screen readers that translate the text to a more appropriate format (e.g., text-to-speech for visually impaired people), the content extraction operation is very valuable. Content extraction (CE) is defined as the process of determining those parts of an HTML document that represent the main textual content [5]. Because different Web pages often have a different layout and a variety of configurations are possible, the task is at first sight not trivial. Recently a number of solutions have been proposed. The problem, however, is to find a solution that is generic (i.e., portable to many types of Web pages), accurate (i.e., find all important content in a precise way) and efficient (often a large number of Web pages are processed).

We designed, implemented and evaluated a content extraction system that satisfies the above requirements. Our method is simple, generic, robust and efficiently computable. The results are comparable or better than state-of-the-art methods that are computationally more complex, when evaluated on a standard dataset used in the literature. The research was done in the frame of a project where we crawl, clean, classify, summarize and index Web pages.

The remainder of the paper is organized as follows. Section 2 discusses related research. In section 3 we present our method for content extraction. Section 4 describes how we evaluate this method, and gives results and a comparison to existing methods. We conclude in section 5 where we also give some hints for future research.

2. RELATED RESEARCH

The simplest way to clean Web pages is to remove metadata and tags from the source data. The derivation is a fast, single-pass process. However, most often a deeper processing is needed in order to extract the main content, because Web data are infiltrated with advertisements and interaction menus. Early approaches to the content extraction problem heavily relied on a priori knowledge of the Web site's layout and formatting [10, 3], knowledge which could eventually automatically be learned, but the approach suggests that only a limited amount of formating templates for Web pages are used, which is an unrealistic assumption.

Gradually interest grew in building generic content extraction systems that operate on all types of Web pages. Usually the main text of a Web page is long and homogeneously formatted, while additional contents are usually highly formatted and contain little and short texts. These and other signaling cues for relevant and irrelevant content were exploited in various ways. [7] starts from the HTML tree and wraps its relevant content as a subtree that contains a large number of visible text elements, and which fans out into many children. [6] use the high ratio of link content to detect navigational menus and similar structures. [11] operate on the DOM (Document Object Model) tree, which defines the logical structure of well-formed HTML or XML documents, and identify hyperlinked clutter as text advertisements and long lists as syndicated references to other structures. [12] detect a continuous part of the document which contains text based on the analysis of so called document slope curves. A document is represented as a binary vector. HTML tags except for the ones that indicate content

(e.g., font changes) are given a weight one, all other tokens are given a weight zero. From this vector a document slope curve is generated. The entries in the document slope curve graph correspond to the total of the binary vector entries up to and including each token. Long, low sloping regions of this graph represent content (text without tags). [13] use the text-to-tag ratio of lines of a document to find clusters of content in a Web page.

The closest to our approach is the Content Code Blurring (CCB) method of [5] that implements several methods to identify those parts of a Web page which contain a lot of text and few or no tags. A document is represented as a sequence of text and tag (code) characters or tokens stored as a binary content code vector. The code vector is blurred by using a Gaussian blurring filter (by iteratively spreading the values of a character or token to its neighbors until the values stabilize), after which the areas with high content bearing values are extracted. A variant of the method, Adapted Content Code Blurring (ACCB), is better suited to wiki style documents and ignores anchor tags. [5] makes a comparison with the methods described by [12] and [6], where he shows that the content code blurring method that ignores the anchor tags outperforms the former methods.

More sophisticated approaches extract the information from the visually rendered output of a Web page or other HTML content. Such an approach was following by [9], who extracted tabular data from rendered pages and [1] who classified emails based on the content that is rendered in the email browser. Although very valuable and generic, especially in an adversary setting where certain content might be present in the source, but hidden for the user of the browser, these approaches are computationally much more expensive than the method for Web page cleaning that we propose.

3. GOALS

We are given a source file containing HTML markup-tags and text. As shown in fig.1, this text consists of some relevant content ("main content") but also a lot of content that is not relevant outside the context of this particular Web page, such as navigational menus, comments, links to related articles and others. The goal of this article is to develop a method that classifies every character in the source file as being relevant or not relevant, and creates an output file that contains only the main content, cleaned from any markup-tags.

4. METHOD

We develop a method that extracts the main content from Web pages. The difficulty of this task differs largely in different settings. First, there is the setting where the Web page is known at development time. Here, the structure of the page can easily be exploited to accurately extract the main content based on simple regular expressions. In the second setting a limited number of Web sites are targeted. Here, it is possible to use automatic learning methods to creating a method that extracts the main content specifically for every Web site (i.e. site wrapping). In the third setting the Web page is unknown at development time and can originate from any Web site. This task is extremely challenging, since fully successful extraction methods need to perform a semantic analysis of both the text and structure of the Web page, which can be further complicated by common mistakes in spelling, miswritten markup-tags and an ill-defined HTML-structure. In this setting it also not possible to perform site wrapping, since the pages do not originate from a common Web site.

Figure 1: Example Web document with the main content marked.

In this paper we propose a method that performs only a very shallow analysis of the Web page. This method does not depend on strong assumptions on the structure or content of the Web page and is fully language independent. The main idea behind our method is that a Web page has both content text (the news item, blog entry, ...) and garbage text (navigational menus, links to other articles, adverts, comments,...), but that the content texts tend to be continuous, long text with little structural markup, and that the garbage text tends to be short texts with a lot of structural markup. We make the following weak assumptions: The first assumption states that the text representing the content is separated from the garbage text with one or more markup-tags. The second assumption states that no garbage text occurs in the main content, e.g. that the main content text is continuous (not taking into account the markup-tags). The third and most important assumption states that the main content of the text contains less structural markup-tags (see below) than the garbage text.

An informal inspection of some targeted Web sites reveals that both assumption 1 and 2 are always satisfied, and these assumptions are also satisfied in our test set (see section 5.1). The third assumption, although intuitively correct, was violated in some cases. In section 5.3 we will discuss in detail when this occurred and the influence of this violation on the content extraction method.

We first locate a subset of markup-tags that modify the structure of the Web page. These tags include, but are not limited to `<p>`,`<table>`,`<BR>`,`<div>`,`<h1>`,`<h2>`,.. and `<li>`[1]. We ignore the tags that do not modify the structure of the Web page, such as `<b>`, `<a href=..>` and `<font ...>` and we also ignore data that is not content-related, such as JavaScripts, style definitions and HTML comments. We then transform the structured HTML page to a linear list

[1]see http://www.w3.org/TR/xhtml1/

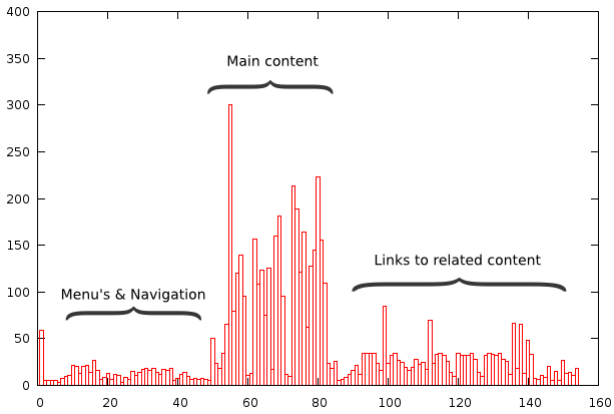


Figure 2: Example plot of the document density

of text strings $L = \{s_1, ..., s_n\}$. We parse the structure of the Web page using a robust HTML parser[2], that will, when presented with a not well-structured HTML page perform a best-effort parse. This parser visits every node in the HTML structure. If a node containing text is encountered, this text is added to the last text string in L. If a markup-tag that modifies the structure of the Web page is encountered, L is extended with one empty string. We continue this process until the entire Web page is parsed.

We build a graphical representation of the array L in fig. 2 where the x-axis represents the position of the array and the y-axis represents the length of the strings at the different positions. In a second step we analyze this graph to find the main content in the Web page. Typically, the main content for a Web page containing news articles is located in the region of L that has the highest density. We therefore convert the problem of extracting the main content of a Web page in the problem of selecting the highest density region of L, for which we have designed a simple algorithm. We first locate the string s_{max} in L with maximum length $maxL$[3]. Then a cutoff length $cutoffL$ is computed as $cutoffL = maxL * c_1$, where c_1 is a constant. We initialize the high density region R as $R = \{s_{max}\}$. We then incrementally add strings s_i to R. A string s_i is added to R iff $length(s_i) > cutoffL$ and there is a string $s_j \in R$ such that $|i - j| < c_2$, where c_2 is a constant. The algorithm terminates when no more strings can be added to R.

To create the final text $t_{automatic}$ containing the main content of the Web page, we find the leftmost string s_l in R and the rightmost string s_r in R. We then create T by concatenating all strings s_i, where i ranges from l to r (inclusive).

Optimal values for c_1 and c_2 were chosen manually when performing the experiments in section 5. The values used in this experiment were $c_1 = 0.333$ and $c_2 = 4$.

Although this algorithm is very simple, it incorporates several interesting ideas. First of all, it does not depend on the structure of any particular Web site, but uses a notion of document density which can be expected to be universal for most Web sites containing news articles. Secondly, it does not depend in any way on the text and is thus fully language independent. Thirdly, it relies only on a limited amount of the HTML-markup, thus making allowances for dirty and non-well structured Web pages.

[2]http://java.sun.com/products/archive/hotjava/
[3]Note that L and thus $maxL$ are page specific

Web site	URL	# of pages	Language
bbc	news.bbc.co.uk	1000	en
chip	www.chip.de	361	de
economist	www.economist.com	250	en
espresso	espresso.replubblica.it	139	it
golem	golem.de	1000	de
heise	www.heise.de	1000	de
manual	different	65	en,de
repubblica	www.replubbica.it	1000	it
slashdot	slashdot.org	364	en
spiegel	www.spiegel.de	1000	de
telepolis	www.telepolis.de	1000	de
wiki	de.wikipedia.org	1000	en
yahoo	news.yahoo.com	1000	en
zdf	www.heute.de	422	de

Table 1: Datasets used for evaluation, showing the name, the URL, the number of pages used in this evaluation and the languages of the datasets.

5. EXPERIMENTS

5.1 Data set used

We evaluate the proposed method on a data set previously used in state-of-the-art content extraction [5]. 14 different datasets (see table 1) were gathered from the Web. A golden standard was created for every HTML page by manually selecting the main content of every Web page. Most Web pages contain news items, although some also contain encyclopedia articles (wiki) or Web pages with different types of contents (manual). The Web sites are written in different languages : English (en), Italian (it) and German (de).

5.2 Evaluation

As described in section 3 we aim at building a method that can successfully label text in a Web page as "main content" or "garbage". In this section we describe how we evaluate the method proposed in this paper.

Although we described the task conceptually as labeling text in a Web page, in reality most (all) systems that perform this extraction task take as input the HTML source code of the Web page and return a file containing the cleaned text. The ground truth data (described above) is also stored in this cleaned text format. To evaluate the developed method, we need a metric that compares how "similar" the automatic output is compared to the manually generated output. More formally, we define t_{manual} as the main content text that was manually created and we define $t_{automatic}$ as the content text that was automatically created.

Different metrics have been proposed that measure the similarity between the two files. In this paper we use two evaluation metrics: longest common substring and longest common subsequence.

Web site	Baseline		Content extraction	
	LCString	LCSequence	LCString	LCSequence
bbc	60.16	61.52	96.32	97.17
chip	6.09	19.25	26.33	78.09
economist	30.91	66.85	45.48	91.88
espresso	69.04	77.32	82.10	89.25
golem	8.28	50.92	15.78	92.17
heise	46.57	61.47	72.28	96.82
manual	11.72	40.72	20.64	53.94
repubblica	14.77	71.95	21.57	90.74
slashdot	10.96	11.61	29.93	53.85
spiegel	8.86	55.86	13.39	86.84
telepolis	5.25	83.14	5.83	89.15
wiki	70.49	81.87	71.96	78.67
yahoo	34.73	65.75	52.36	94.58
zdf	14.39	67.50	25.13	82.93

Table 2: Average results for the baseline and automatic extraction method for LCString and LCSequence evaluation metrics, given in F1-measure (%).

The longest common substring metric (LCString) [2] finds the longest continuous string that appears both in the automatic output and the manual output. For example, the LCString of the strings "the dog jumps over the brown fox" and "the fox jumps over the brown dog" is " jumps over the brown " (of length 20). This metric is useful since it focuses on the longest continuous string, thus highly penalizing any discarded words (or punctuation marks) in the center of the text, which could possible carry high semantic value (e.g. imagine that at some point in the text the word "not" would not be extracted by the automatic method, thus possibly changing the entire meaning of a sentence). On the other hand, a major disadvantage of this method is that it treats all symbols identical, e.g. that discarding a space in the center of the text could possibly half the LCString, thus halving the score on a certain document.

The longest common subsequence metric (LCSequence) [8] finds the longest sequence of characters that appear in that order in both the automatic output and the manual output. For example, the LCSequence of the strings "the dog jumps over the brown fox" and "the fox jumps over the brown dog" is "the jumps over the brown " (of length 23). Notice how LCString is always a substring of LCSequence. This metric is less strict in that it assigns only a modest penalty to missing characters.

We have opted for two character based algorithms (in contrast to for instance [4]), since we feel that word based algorithms are harder to implement (because the characters on a Web page need to be correctly split into words which is not relevant for this task).

For both metrics we calculate, given the length of the longest common string or sequence s_{max}, the familiar information retrieval metrics of precision, recall and F1-measure, as follows:

$$precision = \frac{length(s_{max})}{length(t_{automatic})}$$

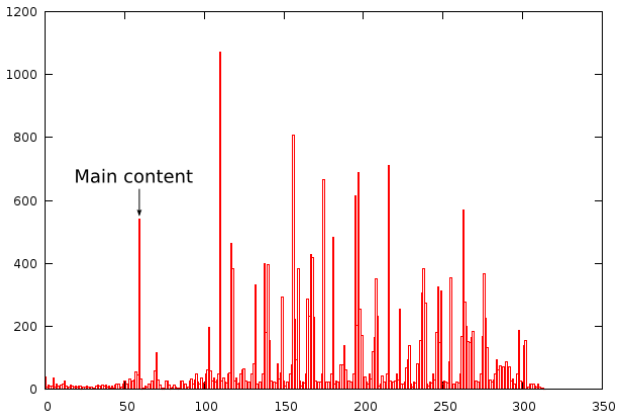


Figure 3: Document density graph for an example of the slashdot corpus.

$$recall = \frac{length(s_{max})}{length(t_{manual})}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

5.3 Results

We perform a first set of experiments where we compare the raw output of the proposed method with the manual extracted texts. This method yielded results that were lower then expected, caused by superfluous spaces in the manual extracted texts. These spaces do not influence in any way the rendering of the page (apart from splitting words) and we thus feel that they can be ignored. All results reported ignore any spaces in both automatic and manual extracted texts.

We compare our approach with a baseline method that extracts *all* texts from a given Web page, removing markup information, but does not perform any content selection. Table 2 shows that the method proposed here results in a significant increase over the baseline. This can be explained by the fact that although the baseline achieves near perfect recall (since the main content will certainly be part of the extracted text), it suffers from a low precision (since it extracts all text). Our method generally also achieves a high recall, but also a high precision because of the dynamic content selection methods.

Table 2 shows that for some datasets (bbc, yahoo, heise) our method achieves near perfect F1-measure. For many other datasets (spiegel, economist, repubblica, espresso, telepolis, golem) our method achieves an F1-measure of 85% or more, which is certainly satisfying given the complexity of the task.

The proposed method achieves disappointing (although better then baseline) results on only two datasets, slashdot and manual. On the slashdot dataset we achieve a LCSequence F1-measure of 53.85%. The reason for this result can easily be explained by fig. 3. We see here that the string representing the main content (as indicated) is only a small subset of the entire text on the Web page. Closer inspection reveals that the remaining text appearing after this content

	Baseline		Content extraction	
Web site	Here	In [5]	Density	ACCB
bbc	61.52	59.5	97.17	92.4
chip	19.25	17.3	78.09	70.3
economist	66.85	61.3	91.88	89.0
espresso	77.32	62.4	89.25	87.5
golem	50.92	50.2	92.17	95.9
heise	61.47	57.5	96.82	91.6
manual	40.72	37.1	53.94	41.9
repubblica	71.95	70.4	90.74	96.8
slashdot	11.61	10.6	53.85	17.7
spiegel	55.86	54.9	86.84	86.1
telepolis	83.14	85.8	89.15	90.8
wiki	81.87	82.3	78.67	68.2
yahoo	65.75	58.2	94.58	73.2
zdf	67.50	51.4	82.93	92.9

Table 3: Average LCSequence F1-measure (in %) results of the density method reported here and the ACCB method reported in [5].

Web site	ACCB	density
bbc	1.0	0.361
chip	8.0	0.314
economist	15.0	0.294
espresso	16.0	0.317
golem	9.0	0.337
heise	12.0	0.341
manual	20.0	0.353
repubblica	14.0	0.355
slashdot	13.0	0.353
spiegel	15.0	0.351
telepolis	52.0	0.377
wiki	28.0	0.377
yahoo	13.0	0.315
zdf	1.0	0.318

Table 4: Average processing time (in s/Mb) for our density extraction method and the ACCB method reported in [5].

are comments. Indeed, a page on the slashdot Web site typically consist of a small text describing some newsworthy fact and many comments. Often the length of an individual comment is larger than the length of the news item. It is thus easy to understand how our method, relying heavily on the text density, fails for these documents.

The result on this dataset shows one of the limits of our approach. Since we assume that the density vector has a shape similar to fig. 2, we expect our method to fail on documents with a the density vector that has a very dissimilar shape. This is also the reason why the results on the wiki Web site are not excellent. We can thus see that it is important to consider the type of the Web pages before employing this technique, although in the course of the Acknowledge project we experienced that most targeted Web sites did not pose a serious problem for successful cleaning.

We found that the reason for the low performance on the manual dataset is not due to the automatic extraction method, but due to the golden standard for that dataset. For many files, the golden standard is not correct and misses large parts or all of the main content for a particular Web site. We are not aware of the reason for this errors, but note that the low result on this dataset in [5] can probably also be attributed to this incorrect golden standard.

It is hard to compare our results with existing work on this task. We are currently only aware of one work that also employs this dataset [5]. The authors evaluate their methods using the LCSequence metric, using a word based algorithm, and not a character based algorithm as reported here. Although this might lead to slightly different results, we feel that these metrics are still close enough to allow for a comparison.

Table 3 compares our content extraction method to the currently best method (Adapted Content Code Blurring) reported in [5]. We also compare the baseline results reported here with our baseline result. Surprisingly, we notice that for some datasets (e.g. espresso, respectively 62.4% and 77.32%) there is a large difference between the two baseline systems, although for other systems this difference is only very modest (e.g. golem, with respectively 50.2% and 50.92%). This difference can only partially be explained for by different evaluation metrics, but must also be caused by a weaker baseline in [5], possibly caused by a less robust HTML parser or by accidentally adding non-textual content (e.g. JavaScripts) to the baseline result.

When comparing our content extract method to the ACCB method, we see that the proposed density method achieves on average 84.00% F1-measure where the ACCB method achieves 78.16%. We achieve significantly higher results for 9 datasets (bbc, chip, economist, espresso, heise, manual, slashdot, wiki and yahoo), comparable results for 1 dataset (spiegel) and significantly lower results for 4 datasets (golem, repubblica, telepolis and zdf). These results might indicate that our method achieves generally better results. This belief is further strengthened by the fact that our method seems more robust, since our method achieves less then 75% F1-measure on only two (slashdot and manual) dataset, compared to 5 datasets (chip, manual, slashdot, wiki and yahoo) in [5].

We feel that the better results are largely due to the method of creating the density vector employed here. Our choice to use only structural tags, and ignore other mark-up tags makes that this vector reflects closer the real structure of the page, and that text elements that are structurally closer together are also closer together in this vector.

Table 4 shows the processing times needed for the different dataset on a 1.66Ghz Intel cpu. We also compare our processing times with the times reported in [5]. This comparison might not be very accurate since we are not aware of the speed of the computer that generated these results, but still indicates that the proposed method performs at least comparable and probably faster then this state-of-the-art method.

6. CONCLUSIONS

We have presented a novel method for content extraction from Web pages, sometimes also referred to as Web page cleaning. This method relies on a single heuristic that the main content of a HTML page has a high density of text characters and low density of structural code. We have shown that this method performs comparable to, or better than state-of-the-art methods. Furthermore, it has the following valuable properties : (1) it is simple, and easy to implement, (2) it is fast, processing up to 3.4Mb of HTML code per second, (3) it runs robustly on dirty or not well-formed HTML code and (4) it does not use the content of the text itself and is thus language-independent

Furthermore, we have proposed to make a distinction between structural and non-structural markup-tags. A comparison with another state-of-the-art method has shown that making this distinction improves results and allows for more robust methods.

Although we have shown that text density is an important heuristic when extracting content from Web pages, it is naive to expect that all Web pages can be successfully cleaned using this heuristic alone. Therefore, in the future more powerful methods will have to be developed. We think that several research directions could prove to be promising: Firstly, methods that perform an analysis of an entire Web site (as compared to a single Web page) could discover the common structure and texts of all pages of a certain Web site. It can be expected that this common structure and texts do not belong to the main content for a particular Web site. Secondly, one could perform an analysis of the text in a Web page, and learn that certain words do (e.g. "written by", "author") or do not (e.g. "close this window", "comments") belong to the main content.

The method as described here works on the raw density values. It might be advantageous however to create a more abstract representation which potentially allows more powerful algorithms. For instance, we could approximate a smooth function such as a polynomial function to the density values using a least squares method. We could then use the maximum, minimum, first and second order derivatives to select the highest density region.

Acknowledgments

We would like to thank Thomas Gottron for providing us the dataset which was used in the evaluation of this work. The work reported in this paper was partially financed by the Acknowledge project (IBBT) and partially by the AMASS++ project (IWT 060051).

7. REFERENCES

[1] A. Bergholz, G. Paass, F. Reichartz, S. Strobel, M.-F. Moens, and B. Witten. Detecting known and new salting tricks in unwanted emails. In *CEAS 2008: Proceedings of the Fifth Conference on Email and Anti-Spam*, 2008.

[2] P. E. Black. "longest common substring". In *Dictionary of Algorithms and Data Structures [online].* http://www.nist.gov/dads/, 2004, retrieved 17/11/2008.

[3] A. Finn, N. Kushmerick, and B. Smyth. Fact or fiction: Content classification for digital libraries. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.

[4] T. Gottron. Evaluating content extraction on HTML documents. In *ITA'07: Proceedings of the 2nd International Conference on Internet Technologies and Applications*, pages 123–132, 2007.

[5] T. Gottron. Content code blurring: A new approach to content extraction. *International Workshop on Database and Expert Systems Applications*, 0:29–33, 2008.

[6] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm. DOM-based content extraction of HTML documents. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 207–214, New York, NY, USA, 2003. ACM.

[7] W. Han, D. Buttler, and C. Pu. Wrapping web data into xml. *SIGMOD Record*, 30(3):33–38, 2001.

[8] D. S. Hirschberg. Algorithms for the longest common subsequence problem. *J. ACM*, 24(4):664–675, 1977.

[9] B. Krüpl, M. Herzog, and W. Gatterbauer. Using visual cues for extraction of tabular data from arbitrary HTML documents. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1000–1001, New York, NY, USA, 2005. ACM.

[10] S.-H. Lin and J.-M. Ho. Discovering informative content blocks from web documents. In *KDD '02: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 588–593, New York, NY, USA, 2002. ACM.

[11] C. Mantratzis, M. Orgun, and S. Cassidy. Separating XHTML content from navigation clutter using DOM-structure block analysis. In *HYPERTEXT '05: Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia*, pages 145–147, New York, NY, USA, 2005. ACM.

[12] D. Pinto, M. Branstein, R. Coleman, W. B. Croft, M. King, W. Li, and X. Wei. Quasm: a system for question answering using semi-structured data. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 46–55, New York, NY, USA, 2002. ACM.

[13] T. Weninger and W. H. Hsu. Text extraction from the web via text-to-tag ratio. *Database and Expert Systems Applications, International Workshop on*, 0:23–28, 2008.

Patent Search: An important new test bed for IR

John Tait, Mihai Lupu
Information Retrieval Facility
Palais Eschenbach
Eschenbachgasse 11/3
1010 Vienna, Austria
{john.tait, m.lupu}
@ir-facility.org

Helmut Berger, Giovanna Roda, Michael Dittenbach, Andreas Pesenhofer
Matrixware
Lehargasse 11/8
1060 Vienna, Austria
{h.berger, g.roda,
m.dittenbach,
a.pesenhofer}
@matrixware.com

Erik Graf,
Keith van Rijsbergen
Dept. of Computing Science
University of Glasgow
Glasgow G12 8QQ
{graf,keith}@dcs.gla.ac.uk

ABSTRACT

The patent system is a major pillar of intellectual property management. In an economic environment that is becoming increasingly knowledge-based, the importance of patents is steadily rising. Processes such as drafting, examining and defending patents depend critically on effective search spanning both the world-wide body of patents and non-patent literature. This paper provides an overview of recent work at the Information Retrieval Facility (IRF) and Matrixware within this domain and is a call to IR researchers and professionals to engage with the challenges of patent search. To provide these grounds, we describe why patent search is an important challenge, why we believe it to become a major focus of IR research in the next few years, and what actions we take to create a new test bed for experimentation.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Systems and Software; K.4.0 [**Computers and Society**]: General

Keywords

Intellectual Property, Patent, Information Retrieval, Test Collections, Evaluation

1. INTRODUCTION

Patent protection is one of the means by which individuals and organizations protect and exploit intellectual property.

In a 2005 study[1], the European Commission estimated the median value of a patent to be 300K Euros with 10% of patent owners reporting values of 10 million Euros or more. Over 50 million existing patents around the world and ever rising patent filing rates (1.76 million in 2006 worldwide; increase of 4.9% over 2005[2]) indicate the enormous economic value of patents and evidence the need for effective and efficient patent search tools. Industries with major interests in patents range from the biomedical and pharmaceutical to the automotive, aerospace, information and communication technologies sectors [12, 24]. Protecting intellectual property rights through patents is crucial not only for large enterprises maintaining large IP portfolios, but also for small and medium-sized enterprises (SMEs) in need to secure the economic value of their investments in research and development. SMEs are especially in demand of affordable means to protect their Intellectual Property from large corporations and "Patent Trolls", i.e. small, non-competing *Patent Licensing and Enforcement Companies (P-LECs)*.

Highlighting another substantial economic factor of patents, the European Patent Office estimates that the *'European industry is losing US$20 billion every year due to lack of patent information, which results in duplication of effort such as reinventing existing inventions, resolving problems that have already been solved, and redeveloping products that already are on the market'* [16].

A patent is a bargain between the inventor and the state. The inventor must teach the community how to make the product and use the techniques he has invented in return for a limited monopoly, which gives him a set time to exploit his invention and realize its value. Besides being important legal documents, much knowledge never published elsewhere is distilled through patents, rendering them a valuable technical resource. In the 21st century, as most patents are filed in digital form and older patents being digitized using optical character recognition (OCR) technologies, patent data is increasingly accessible through the Internet.

To be valid and legally defensible, a patent must not attempt to patent an invention for which there is prior art; i.e., the invention has either been previously patented, or publicly disclosed, for example by appearing in a scientific journal. This gives rise to a major secondary industry –

[1] `http://ec.europa.eu/internal_market/indprop/docs/patent/studies/patentstudy-report_en.pdf`

[2] `http://www.wipo.int/ipstats/en/statistics/patents/wipo_pub_931.html`

DIR 2009 Enschede, The Netherlands

patent search – which provides tools to patent agents, patent attorneys and various other classes of end users.

Matrixware's commitment to improve existing patent search technologies led it to engage in several commercial and non-commercial programs, some of them carried out in cooperation with the IRF and its academic partners. Alexandria and Leonardo are the two products of the patent search product portfolio being currently developed at Matrixware. Some effort has also been invested in the area of evaluation, with the establishment of two tracks for the intellectual property domain within the CLEF and TREC evaluation campaigns.

The remainder of the paper is structured as follows. Section 2 states the problem of patent search in general. The topic of IR evaluation and two particular evaluation tracks are described in Section 3. Current patent search systems in industry and academia are outlined in Section 4. A description of a comprehensive patent repository and an application framework supporting the research and development of IR software and services is given in Section 5. We conclude with a call for participation in the presented activities and invite IR experts to take up research in the challenging field of patent search in Section 6.

2. THE PATENT SEARCH PROBLEM

Professional patent search, as done by patent attorneys, independent or corporate patent searchers, and patent office examiners, is a highly specialized form of information search. In fact, there are different search types in the patent domain. Atkinson [3] states that *'depending on what group is doing the asking, the types of patent search requested may include simple patentability, clearance to market a product, validity, opposition to a patent being sought by another, infringement watch, creating IP landscapes for business development or R&D, infringement defense, litigation, prosecution support, and creation of portfolios for assignments, investments, mergers and acquisitions, and licenses with legal status and contingency clauses'.*

Patent searches are different in a number of ways to the operation of Internet search engines. Searches tend to be based on lengthy search sessions rather than single queries: results review and query refinement may take several days of work. Very high recall is required: not a single relevant document should be missed by the search. This contrasts with Web search where the aim is high precision: few or no irrelevant documents shall appear among the top ranking Web pages. Searchers often need to review a very large number of results to gain an overall view of the existing patent landscape in a field, or to identify opportunities for new patents in gaps in the prior art. Missing even a single relevant document can have severe financial and economic impact for a company when, for example, a granted patent becomes invalidated because of a document that has not been retrieved at application time.

Recently however, there is a growing lack of confidence in the patent system partly due to the difficulties faced by the community of patent examiners, agents, and searchers to locate all relevant documents. The following reasons are identified to be mainly responsible for this development: i) increase in volume, but also ii) the obscurity of patent language, iii) the complexity of technologies and iv) the problems of classifications in keeping pace with technological developments. Finally, the newest technologies for Information Retrieval are not fully integrated in tools for patent search.

In addition to these requirements the patent documents are complex in language and structure. Amongst others, they contain a description of the invention's background similar to scientific literature and the claims section using a very distinct legal language to define the scope of the protected technology. Multilinguality, references to other patent and non-patent documents as well as images, DNA sequences and chemical formulae contribute to the complexity of the patent structure.

We refer to [22] for some more interesting remarks on what is peculiar about patent data and why it will contribute to IR research. A description of the technical and legal aspects involved in patent search is provided in [1] and [6].

3. BUILDING A MODERN IR TEST BED

Since their introduction, test collections in Information Retrieval have played a pivotal role in the evaluation of retrieval models [25, 27, 32]. One of the first test collections was defined as part of the Cranfield Experiments [5] and provided the blueprint for subsequent IR test collections. The *'significant achievement of Cranfield 2 was to define a notion of the methodology of IR experimentation'* [25]. A design goal for the Cranfield 2 experiment was to create a laboratory type situation by reducing the number of operational variables during experiments. To allow for this measurement of information retrieval effectiveness in a standardized way, a test collection has to consist of three elements: A document collection (corpus), a task represented by a suite of specified information needs (topics), and a set of relevance judgments associated with the topics. Although not unchallenged [20], this approach has found widespread adoption in IR, and can nowadays be seen as the standard system evaluation method. Existing collections have enabled IR research to be conducted on the retrieval of news stories, Web pages and government documents. Other areas actively being explored are blogs and enterprise documents.

The creation of test collections in the patent domain has been pioneered by the NII Test Collections for IR Systems project (NTCIR)[3]. First introduced in the NTCIR-3 workshop (2001-2003), the patent task has led to the release of several patent test collections. Details of these collections are provided in Table 1. These test collections, primarily targeting Japanese patent documents, have been associated with a variety of different user tasks:

1. Cross-language, cross-genre retrieval (NTCIR-3 [19]): Given Japanese, English, and Chinese newspaper articles associated with a particular technology or commercial product, the task consisted of retrieving Japanese patents relevant to the article. The relevance assessments for this task were created manually.

2. Associative retrieval (NTCIR-3 [19]): The task consists of retrieving patents for a given search topic (i.e. either a newspaper article or patent). Participants were asked to submit a list of retrieved patents and passages associated with the topic. Again, the relevance assessments were created manually.

3. Invalidity search (NTCIR-4, -5, and -6; see [9, 10, 11] respectively): Participants were asked to search a target patent collection for patents that can invalidate

[3] `http://research.nii.ac.jp/ntcir/`

Workshop	Document Type	Time Period	# of Docs.	# of Topics	Rel. A. Creation
NTCIR-3	Patent JPO(J)	1998-1999	697,262	31	Manual
	Abstracts(E/J)	1995-1999	ca. 1,700,000	31	Manual
NTCIR-4	Patent JPO(J), Abstracts(E)	1993-1997	1,700,000	103	Manual
NTCIR-5	Patent JPO(J), Abstracts(E)	1993-2002	3,496,252	1,223	Inferred
NTCIR-6	Patent USPTO(E)	1993-2002	1,315,470	3,221	Inferred

Table 1: Overview of NTCIR patent test collections (E=English, J=Japanese)

the demand in a given claim. In practice, for each search topic (i.e. a claim), each group submits a list of retrieved patents and passages associated with the topic. The task aimed at identifying patents that can invalidate a topic claim by themselves (1) or in combination with other patents (2).

4. Patent classification (NTCIR-5 and -6; see [17, 18] respectively): The purpose of this task lay in categorizing target patent applications based on the F-term classification system. A submission consisted of a ranked list of F-term classification codes for each target patent application.

These tasks and collections provide a significant step towards patent retrieval specifically in Japanese, and outline the variety of evident user tasks in the patent domain. They also shed light on the specific challenges associated with the creation of test collections in the patent domain. The patent domain provides a distinctly different set of research problems. One of the reasons for the lack of patent retrieval test collections stems from the complexity introduced by the dual nature of patents. Patents are devised as means of intellectual property protection and exhibit both informative and judicial characteristics. Determining the relevance of a patent document is therefore a task requiring legal as well as subject expertise. This renders the task of creating relevance assessments significantly more difficult than in other domains, such as Web pages or news stories. It requires an evaluation of the feasibility of established techniques related to relevance assessment, such as pooling.

Induced by these difficulties of performing manual assessments as done in NTCIR-3 and -4, the creation of the NTCIR-5 and -6 test collections relied upon inferring relevance assessments from references found in patent documents (see the rightmost column of Table 1). Additionally, the question arises which measures would be most suitable for the evaluation of tasks in the patent domain. As mentioned before, patent-related tasks are recall-focused unlike most precision-oriented tasks in the Web domain. The challenges of developing new measures emphasising recall are shared by the legal domain. A discussion of potential measures and the difficulties associated with estimating recall are detailed in the proceedings of the TREC legal track [31]. In light of the variety of tasks outlined above, specific challenges stemming from the vast range of covered technological domains and the significant differences of patents issued by different issuing authorities, the establishment of new test collections for the patent domain will form a necessary step towards advancing information access in the patent domain. In the following we provide an overview of the CLEF-IP and the TREC Chemistry Track, two new initiatives that address specific challenges and opportunities associated with the creation of new testbeds for the patent domain.

3.1 CLEF-Intellectual Property Initiative

The lack of an IR benchmarking suite for European patents gave rise to the CLEF-IP initiative in 2008. This initiative forms part of the Cross-Language Evaluation Forum[4], which promotes research in multilingual information access. The CLEF-IP track (starting in 2009) aims at enabling patent retrieval research in a multilingual environment such as the one presented by the body of patent issued by the European Patent Office. The track is coordinated by the IRF in cooperation with Matrixware, who will provide a large repository of annotated patent data. The tasks will focus on prior art search, i.e. identifying all information that might be relevant to a patent's claim of novelty. Forming a central part of a patent's drafting, grant, and opposition procedures, the search for prior art is one of the most commonly executed tasks, and in light of the rapid growth rate of patent applications, it forms one of the most critical activities in the patent domain. The planned test suite will be multilingual and comprise more than one million patent documents in English, German, and French.

The main bottleneck in the creation of test suites for Information Retrieval is rooted in the labor requirements associated with assessing the relevance of large sets of documents. This is particularly true for patents, where the assessment of relevance requires technological as well as judicial expertise. As stated in the previous section, efforts for creating patent test collections have led to the technique of inferring relevance assessments from the references found within patents [10], because of the high costs associated with manually created assessments. The idea of building a test collection automatically by using citations has been around for some time and has been applied to several domains. In the following we will shortly outline the feasibility of this approach in the patent domain. For a more detailed report on this topic, we refer to [13].

At the European Patent Office, the task of identifying relevant prior art is conducted for each newly filed patent application. Whenever an inventor files a patent application, a patent examiner will conduct such a search and issue a search report listing the relevant prior art, based on the judicial regulations and the examination guidelines. An excerpt from the European Patent Convention – *Implementing Regulations to the Convention on the Grant of European Patents, Rule 61* – outlines the role of the search report [8]:

> *"The European search report shall mention those documents, available to the European Patent Office at the time of drawing up the report, which may be taken into consideration in deciding whether the invention to which the European patent ap-*

[4] `http://www.clef-campaign.org/`

plication relates is new and involves an inventive step."

This report is then filed together with the patent application. The documents mentioned in the search report are called references. In addition to the search report, references may also appear in the patent document itself and in some cases in legal files arising from opposition procedures. While these references are not limited to patents but may include any other type of document where information relevant to the invention's claim of novelty might have been published, initially only patent documents are going to be considered for the task. CLEF-IP is going to utilize this information that is already available in European patent databases for creating its prior art test suite. Since references are labeled with respect to their level of relevance on search reports, relevance assessments will be on a ternary scale (*very relevant*, *relevant*, *not relevant*). This shall lead to a more fine-grained evaluation than the usual one based on binary relevance judgements. The test collection, being generated in an almost completely automatic fashion, is going to include a large number of topics. Hence providing a realistic, experimental test bed.

Through the creation of the CLEF-IP test collection we aim at bringing about new insights on effective methods of performing cross-language retrieval in a patent environment, and how to automate the process of prior art search, for instance by suggesting optimal methods for formulating a query based on the text of a patent application.

3.2 TREC Chemistry Track

In 2009, the Text REtrieval Conference (TREC) will organize a Chemistry Track to evaluate the performance of existing text retrieval tools on a corpus of documents containing chemical formulae, reactions and drawings. Like the CLEF-IP collection, it will contain a large set of patent data. It will build on the lessons learned in the TREC Legal Track and Genomics Track, but it also has it's own challenges. For instance, compared with the Legal Track, it is similar in the fact that high recall is preferred to high precision, due to its focus on patent data and queries, but it differs in the number of documents that are actually relevant for a query (from potentially several hundreds in the Legal Track to only a few tens at most in the Chemical Track). Compared to the Genomics Track, it is similar in the significant importance of special words (genes, proteins versus chemical names), but it differs in the variability of expressing those names (there is less standardization across different types of chemistry: bio-chemistry, polymers, plastics, etc.).

For the TREC Chemistry Track test collection, the main difference from the CLEF-IP Track test collection, apart from the obvious focus on chemistry, is the fact that it contains not only patent data, but also scientific articles. In fact, the collection currently contains over 100,000 patent documents and 45,000 scientific articles from the Royal Society of Chemistry. This inclusion is vital for the success of the track because, there are many references in patents that cite scientific articles as prior art in such specific fields. The scientific articles will be available in XML format, and, like in the patent files, chemical entities will not be identified with special tags. However, a participant might take advantage in tagging them by the layout tags that exist. For instance, a formula like H_2O may appear as *H<inf>2</inf>O*. However, considerable challenges still remain and we will be looking at how much better systems using entity discovery mechanisms are than those doing plain text retrieval.

The queries for this test collection, though still under discussion, will be generated based on patent data, specifically patents that have been opposed by a competitor in the industry. Such references, introduced by the competitor, are generally the most accurate ones, due to the obvious direct interest it has. However, because of the nature of the data, we will also work with volunteers to help us identify those cases that have been judged very relevant by the retrieval systems, but do not necessarily appear in the citations within the patent dossier, i.e. the patent itself, the Patent Office's search report and the opposition references.

4. PATENT RETRIEVAL SYSTEMS

4.1 Patent Offices' Search Systems

The patent offices are obliged to publish new applications, usually together with the search reports and the publications cited in the patent text. This information is mainly provided by data feeds to the public. The patent offices also offer Internet-based services for searching their patent collections.

Espacenet[5], a free search service provided by the European Patent Office (EPO), allows access to the documents filed at EPO as well as at the World Intellectual Property Organization (WIPO). However, the system is quite limited in terms of the search options it offers. For instance, it only indexes terms appearing in the title or the abstract.

Another system provided by the EPO is *Epoline*[6], which allows searching on a few more fields (most notably "opponents") and is frequently used by IP specialists, though it does not index the full-text content of the patents.

PATENTSCOPE[7], WIPO's service for searching international patent applications, allows the use to formulate Boolean and near queries on different sections (e.g. front page, title, abstract, claims) of the patent document.

The United States Patent and Trademark Office (USPTO) also offers a free full-text search service fot US patents[8].

4.2 Commercial Providers

The major commercial player in this field is Thomson Reuters[9] having a relatively new product (*Thomson Innovation*) on the market that combines intellectual property, scientific literature, business data and news with analytics, collaboration and alerting tools in an integrated platform. Other products of the company are *Aureka*, an IP management and analysis platform; *Delphion*, a patent research, analysis and visualization tool and *Derwent World Patents Index*, a comprehensive database of enhanced patent documents.

Chemical Abstracts Service[10] provides their customers products and service relevant to chemistry, plus a wealth of information in the life sciences and a wide range of other scientific disciplines. Advanced search and analysis technologies deliver the most complete, cross linked and effective digital information environment for scientific research and discov-

[5] `http://ep.espacenet.com/`
[6] `http://www.epoline.org/`
[7] `http://www.wipo.int/pctdb/en/`
[8] `http://patft.uspto.gov/`
[9] `http://www.thomsonreuters.com/`
[10] `http://www.cas.org/`

ery. Their main product lines are *SciFinder*, *STN*, *STN Express*, and *STN AnaVist*.

PatentCafe[11] is promoted to be the industry's only international latent semantic analysis patent information research solution with advanced linguistic search capabilities. Their model, or part of it, seems to be described in [26], as one of the authors come from this company. [26] shows how latent semantic indexing (LSI) can be used in patent searching to overcome the drawbacks of Boolean searching and produce a more accurate retrieval result. However, the paper lacks a detailed comparison of the proposed method with the Boolean and the vector space model.

Digital libraries like PubMed[12], Elsevier[13], ScienceDirect[14], ACM Digital Library[15], and IEEE Xplore[16] can be seen as important sources of information for searching scientific, non-patent literature. However, legally, any publication, wherever it has appeared, can be considered as prior art.

An actual example of how such systems are used for literature and patent citations search with a subsequent analysis of the search results is given by Badger in [4].

4.3 Corporate Information Centers

The amount of services larger comapnies need to use has led to the development of Corporate Information Centers, i.e. a single point of information access. Corporate Information Centers provide managers, researchers and scientists with easy access to company-related information as well as access to the wealth of knowledge beyond the company's intellectual property. Corporate Information Centers are concerned with the fast delivery of the i) right information ii) at the right time, iii) in the right context and iv) with the required quality.

Corporate Information Centers may further assist users by linking, for example, data from one database vendor to the data provided by another highly specialized supplier. The right configuration including all necessary content and information services paired with the personnel's attitude towards this system may result in a high-performing asset for the organization.

Such Corporate Information Centers can be implemented as simple Web pages listing all links to the subscribed information services, or in a more sophisticated way, as a rich intranet site making all services accessible in a unified fashion may exist. At Henkel, the Henkel Info Center serves i) as single search system for the intranet and the Internet, ii) as visualization framework of search results and iii) as integration and linking instrument of available sources of information on the intranet and Internet [14].

Another approach is to distribute a rich client application that has search plug-ins for connecting to the content providers included to all workstations (c.f. Section 5.2).

4.4 Related work in academia

Though there are no patent retrieval systems that are a direct consequence of any one academic research project, it is worth mentioning here that the challenges of patent retrieval have triggered the interest of information retrieval academics as well. Probably a defining moment was the SIGIR Workshop on Patent Retrieval [21] in 2000. After that, it would seem that research in this area was carried out mostly in Asian academic centers. In [15], Higuchi et al. present a retrieval system for cross lingual (Japanese/English) patent retrieval. Takaki et al. suggest a solution to finding similar patents to a given one by automatically splitting the patent file into a set of sub-queries and then aggregating the results [30].

The goal of PATExpert [33], an advanced patent processing service, is to push forward the adoption of the semantic paradigm for patent processing and to provide a user technique allowing for more powerful access to the content of textual patent documents. A content representation schema for patent documentation and sketch the design of techniques that facilitate the integration of this schema into the patent processing cycle is introduced. On the one hand, these are techniques that facilitate the access to the content of patent documentation provided in a textual format – be it by the human reader or by the machine – in that they rephrase and summarize the documentation and map it onto a formal semantic representation. On the other hand, the second type of techniques operate on the content representation.

5. CREATING SOFTWARE ENVIRONMENTS

5.1 Alexandria

One major objective of Matrixware is to develop a comprehensive patent document repository and factory, Alexandria. This repository aims at i) providing high-quality, standardized first-level patent literature and ii) enriching first-level patent literature by integrating a plethora of sources of evidence. In a nutshell, Alexandria is an ever-growing content repository. It is designed as an extensible, global storage facility for high quality scientific, technical, and business information. As such, Alexandria is intended to become an authoritative information source providing data services to global R&D communities. The initial offering will primarily cover patent documents. Alexandria will deliver an infrastructure capable of supporting the diversity of data services required by professional researchers, information specialists, and software engineers. More specifically, access to the repository is provided either natively or via subscription to data feeds.

Taking advantage of collaborative patent annotation tools, community generated meta-data will be integrated with the patent document repository and made accessible through the enhanced version of Alexandria. Personalized subsets of this repository can be compiled at will for specific research or commercial purposes. It is envisaged that a mix of public domain or royalty-free content as well as content that may attract royalties for source publishers will find inclusion in the offering.

5.1.1 Processing

From a high-level view, incoming raw data originating from diverse sources is transformed according to an XML schema, which, among other harmonization steps, unifies the incoming documents. Then standardization and normalization of the data is carried out. To achieve high-quality, in-depth quality assurance is applied on the data. This includes the analysis of gaps in coverage and the content of

[11] `http://www.patentcafe.com/`
[12] `http://www.ncbi.nlm.nih.gov/pubmed/`
[13] `http://www.elsevier.com/`
[14] `http://www.sciencedirect.com/`
[15] `http://portal.acm.org/`
[16] `http://ieeexplore.ieee.org/`

fields. As an example consider the identification of missing patent numbers, correction of missing country codes in citations, missing IPC classes, misclassification, missing inventor or assignee names, to name but a few. Additionally, all changes and operations on data in the repository is captured and stored allowing on-demand revision history and tracking of, for instance, classification changes, re-assignments, changes in the legal status, etc.

5.1.2 Coverage

At the time of writing, the Alexandria repository contains XML files covering EP (European) applications and granted patents until December 2005 (sums up to about 1.2 million EP documents). From a technological perspective, the processed data sources are stored in an SQL database that is not available to the end users (yet). However, Alexandria is set up and ready for personalized data extracts. Currently the following data sources are included:

- DocDB data. Bibliographic data, abstracts, citations and family from more than 70 authorities.
- INPADOC status. Legal status information from more than 40 authorities. This information is not included in the current data extracts but will be available soon.
- EP A and EP B (applications as well as granted patents) before 2006. Bibliographic data, original abstracts, English translation of German and French abstracts, description and claims.
- WO (WIPO) OCR full text for PCT[17] patents published in English, French and German until Nov. 2008.
- US applications and granted patents before 2005.
- Patent Abstracts of Japan (PAJ) until Nov. 2008.
- Austrian, Swiss and Belgian OCR full-text until Nov. 2008.

5.2 Leonardo

5.2.1 Overview

The Leonardo Framework is an open-source application framework that supports research and development of information retrieval software and services (see Figure 1). The Leonardo Workbench, as the central part of the framework, is a community-based development environment fostering technology transfer between intellectual property professionals and information retrieval experts. The primary objective of this environment is to provide the users with a comprehensive library of tools that enables users to access and take advantage of repositories such as Alexandria and other external information resources.

In the basic configuration, the Leonardo Framework serves two user groups: first, the intellectual property professional and second, the information retrieval expert. Henceforth, these user groups are referred to as the IP/IR community. For the information retrieval expert the Leonardo Framework is a comprehensive and reliable integrated development environment. For the intellectual property professional (e.g. a patent researcher), it is a solid interface to a variety of data sources and information processing.

[17]PCT stands for Patent Cooperation Treaty and is a legal framework for filing patents in all treaty member states simultaneously (international patent application).

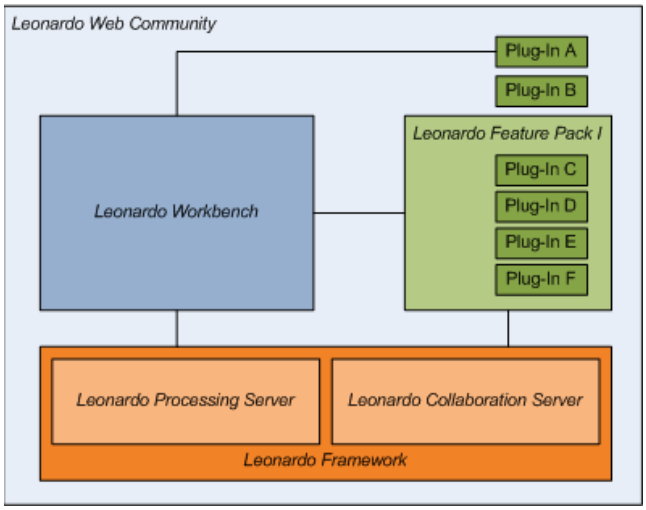


Figure 1: Leonardo components

5.2.2 Technological Building Blocks

The Leonardo Framework is the architectural concept and the basic implementation thereof. It allows functional software components to integrate with Leonardo. This includes APIs, basic server components, connectors, process flow control, commercial context, open interfaces and the respective documentation.

Leonardo Workbench.

The Leonardo Workbench is created within the Eclipse framework. It serves as the central tool for the IP/IR community, be it information retrieval experts or vendors of specific functionalities such as graphical reports, data connectors or advanced IR algorithms. It is based on components exported by the Leonardo Framework making use of the Leonardo Collaboration Server as well as the Leonardo Processing Server.

Leonardo Plug-Ins and Feature Packs.

Leonardo Plug-Ins encapsulate functionalities for data processing and interfacing with the user by utilizing the Leonardo Framework and integrating with the Leonardo Workbench. The distinction between Plug-Ins and Feature Packs is a matter of scope. Plug-Ins refer to small entities with limited functionality, while Feature Packs bundle a set of Plug-Ins that aim at the accomplishment of complex tasks within a broader scope. Typically, Plug-Ins and Feature Packs integrate functionality that is not available in the standard Leonardo setting and, thus, provide solutions for specific user needs.

Leonardo Processing Server.

The Leonardo Processing Server is the core of the Leonardo Framework. It provides, but is not limited to, the following functionality:

- The Leonardo Framework Support defines and offers interfaces between client components of the Leonardo Framework and exports basic concepts and components to the Leonardo Workbench as well as other tools.

- Leonardo Framework Integration enables third-party vendors to attach their own systems to Leonardo.
- The Leonardo Upscaling support enables users to execute Leonardo Plug-Ins and Feature Packs on a high-performance infrastructure provided by Matrixware. So, user tasks may be scaled up from client side processing to remote, server-side execution.

Leonardo Collaboration Server.

The Leonardo Collaboration Server allows for cooperation between and sharing of arbitrary Leonardo Workbenches. The Leonardo Group and User Management allows for defining workgroups and users within respective workgroups. The Leonardo Data Repository enables the user to store,retrieve and share data between Leonardo users.

Leonardo Community Web Platform.

The Leonardo Community Web Platform currently being establish around Leonardo allows IP professionals and IR experts to take advantage of the knowledge and expertise of IP/IR Community members. It encourages users to request new extensions to the Leonardo Framework and the implementation of new functionalities. It provides access to discussion panels, special interest groups, a market place, documentation, and the like.

6. PROMOTING TAKE UP OF RESEARCH

Patents already are common subjects of study in scientific areas such as economics [2, 7], scientometrics [28, 34], and law [23], but not that much in information retrieval. Patent databases and their search tools primarily relying on Boolean Logic have more or less bypassed the last two decades of developments and advances in IR research. Consequently, patent retrieval is an interesting and important open challenge on which to do research for several reasons:

1. Search scenarios are often complex and rich: involving many cycles of query formulation; results review and query reformulation over lengthy periods (not infrequently days).
2. The patent system is international and multilingual by its nature: a patent filed today in China in Chinese can be rejected (or at least should be rejected) because of a patent filed in the US 50 years ago in English.
3. The search scenario is frequently collaborative in the round: if a Dutch patent searcher conducting a freedom to operate search in chemical engineering finds a Japanese patent which may be relevant based on its English abstract, it is likely that at least a Japanese translator and a chemical engineer will be needed to determine the patent's relevance. Ultimately many others, such as patent attorneys, commercial managers, scientists, may be involved.
4. Many forms of patent search are high recall: a single missed patent or paper can cause, for example, a invalid patent to be granted.

All this contrasts with Web searching, which has been the focus of so much research in recent years, where the emphasis is on high precision search using short, simple queries with little reformulation or requerying.

Further, in terms of scale, Web search is actually a subset of the patent search problem. Any form of public disclosure of the invention prior to the filing date of the patent can invalidate the patent. The forms of prior publication, which have been used successfully to invalidate patents, go way beyond patents and academic publication. In one notorious case a Disney cartoon was used to argue lack of originality of the idea in a patent. Thus, there is no practical way to limit the information which should be indexed for patent searching.

Successful IR research conventionally depends on three elements:

1. The availability of test collections: corpora from which to retrieve document, sets of queries and corresponding relevance judgements;
2. Access to suitable software frameworks that allow for quick implementation of algorithms for specific IR problems.
3. Access to sufficiently powerful hardware, so that large scale experiments can be run in a reasonable time without excessive effort on tuning the software.

The IRF, supported by Matrixware, is providing all three of these.

The IRF is already acting as a redistributor of TREC test collections for research purposes and is planning to do the same with CLEF collections in the future. In addition to that, new test collections are currently being created as an outcome of the tracks initiated by the IRF within the TREC and CLEF evaluation workshops. Through the TREC-Chem and CLEF-IP tracks (described in Section 3), the IRF is going to produce large test collections dedicated to patents. The target data used for these collections will be extracted from the Alexandria repository. In addition to its engagement in the area of evaluation, the IRF is cooperating with other academic institutions in order to obtain public funding in areas like: interactive information retrieval and log analysis, new approaches to multilingual ontology mining and machine translation.

We provide standard IR packages like Lemur/INDRI and Terrier as well as new frameworks based on Alexandria. We give access to powerful packages like StarP[18] which allows direct manipulation of large matrices, enabling researchers to work directly with matrix-centric rather than database-centric interpretations of IR problems.

Finally of course, we provide free access for university and not-for-profit researchers to our SGI Altix 4700. We are committed to expanding this systems as and when necessary. It is currently at <1% of the maximum configuration in terms of main memory and the number of CPUs installed.

Together with Matrixware, the IRF is promoting and supporting research through other mechanisms. The IRF and Matrixware have sponsored many international conferences including SIGIR, CIKM, ECIR, and Future Internet Systems. They provided support for students to attend the AND08 Workshop on noisy text data at SIGIR 2008, and is supporting several research students. A major activity of

[18] `http://www.interactivesupercomputing.com/products/`

Matrixware and the IRF is the organization of the annual Information Retrieval Facility Symposium in Vienna, bringing together IR experts and IP professionals to foster the knowledge exchange between these two groups [29]. More such activities will be promoted in future.

7. REFERENCES

[1] Adams, S. R. *Information Sources in Patents*. Saur, 2006.

[2] Aghion, P., Bloom, N., Blundell, R., Griffith, R., and Howitt, P. Competition and Innovation: An Inverted-U Relationship. *The Quarterly Journal of Economics 120*, 2 (May 2005), 701–728.

[3] Atkinson, K. H. Toward a more rational patent search paradigm. In *PaIR'08: Proceedings of the 1st ACM Workshop on Patent Information Retrieval* (2008), pp. 37–40.

[4] Badger, E. Techniques for analyzing literature search results. *World Patent Information 30*, 4 (2008), 326–334.

[5] Cleverdon, C. *Readings in information retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, ch. The Cranfield tests on index language devices, pp. 47–59.

[6] David Hunt, Long Nguyen, M. R. *Patent Searching: Tools & Techniques*. Wiley, Hoboken, February 2007.

[7] Eaton, J., and Kortum, S. Trade in ideas patenting and productivity in the OECD. *Journal of International Economics 40*, 3-4 (May 1996), 251–278.

[8] European Patent Office (EPO). *Guidelines for Examination in the European Patent Office*, December 2007.

[9] Fujii, A., Iwayama, M., and Kando, N. Overview of patent retrieval task at NTCIR-4. In *Proceedings of NTCIR-4 Workshop Meeting* (2004).

[10] Fujii, A., Iwayama, M., and Kando, N. Overview of patent retrieval task at NTCIR-5. In *Proceedings of NTCIR-5 Workshop Meeting* (2005).

[11] Fujii, A., Iwayama, M., and Kando, N. Overview of the Patent Retrieval Task at the NTCIR-6 Workshop. In *Proceedings of NTCIR-6 Workshop Meeting* (2007), pp. 359–365.

[12] Gambardella, Alfonso, Harhoff, Dietmar, Verspagen, and Bart. The value of European patents. *European Management Review 5*, 2 (2008), 69–84.

[13] Graf, E., and Azzopardi, L. A methodology for building a patent test collection for prior art search. In *Proceedings of the Second International Workshop on Evaluating Information Access (EVIA)* (2008).

[14] Haxel, C. Patent information at Henkel: From documentation and information to collaborative information commerce. *World Patent Information 24*, 1 (2002), 25–30.

[15] Higuchi, S., Fukui, M., Fujii, A., and Ishikawa, T. PRIME: A System for Multi-lingual Patent Retrieval. In *Proceedings of MT Summit VIII* (2001), pp. 163–167.

[16] Idris, K. Intellectual property: A power tool for economic growth. Tech. Rep. Publication N0 888, ISBN 92-805-1113-0, WIPO, Geneva, 2003.

[17] Iwayama, M., Fujii, A., and Kando, N. Overview of Classification Subtask at NTCIR-5 Patent Retrieval Task. In *Proceedings of NTCIR-5 Workshop Meeting* (2005).

[18] Iwayama, M., Fujii, A., and Kando, N. Overview of Classification Subtask at NTCIR-6 Patent Retrieval Task. In *Proceedings of NTCIR-6 Workshop Meeting* (2007), pp. 366–372.

[19] Iwayama, M., Fujii, A., Kando, N., and Takano, A. Overview of patent retrieval task at NTCIR-3. In *Proceedings of NTCIR-3 Workshop Meeting* (2002).

[20] Jones, K. S. *Information Retrieval Experiment*. Butterworths, London, 1981, ch. 13, pp. 256–284.

[21] Kando, N., and Leong, M.-K. Workshop on patent retrieval SIGIR 2000 workshop report. *SIGIR Forum 34*, 1 (2000), 28–30.

[22] Leong, M.-K. Patent data for ir research and evaluation. In *NTCIR 2: Proceedings of the Second NTCIR Workshop* (2001).

[23] Lerner, J. The importance of patent scope: An empirical analysis. *RAND Journal of Economics 25*, 2 (Summer 1994), 319–333.

[24] Rivettte, K. G., and Kline, D. *Rembrandts in the Attic: Unlocking the Hidden Value of Patents*. Harvard Business School Press, Boston, MA, USA, 2000.

[25] Robertson, S. On the history of evaluation in IR. *Journal of Information Science 34*, 4 (2008), 439–456.

[26] Ryley, J. F., Saffer, J., and Gibbs, A. Advanced document retrieval techniques for patent research. *World Patent Information 30*, 3 (2008), 238–243.

[27] Saracevic, T. Evaluation of evaluation in information retrieval. In *In Proceedings of SIGIR 1995* (New York, NY, USA, 1995), ACM, pp. 138–146.

[28] Small, H. Tracking and predicting growth areas in science. *Scientometrics 68*, 3 (Dec. 2006), 595–610.

[29] Tait, J. Information Retrieval Facility Symposium in Vienna. *SIGIR Forum 42*, 1 (2008), 67–67.

[30] Takaki, T., Fujii, A., and Ishikawa, T. Associative document retrieval by query subtopic analysis and its application to invalidity patent search. In *In Proceedings of CIKM 2004* (New York, NY, USA, 2004), ACM, pp. 399–405.

[31] Tomlinson, S., Oard, D. W., Baron, J. R., and Thompson, P. Overview of the TREC 2007 Legal Track. In *In Proceedings of TREC 2007* (November 2007), NIST. `http://trec.nist.gov/`.

[32] Voorhees, E. M. *TREC: Experiment and Evaluation in Information Retrieval*. Digital Libraries and Electronic Publishing. MIT Press, September 2005.

[33] Wanner, L., Baeza-Yates, R., Brügmann, S., Codina, J., Diallo, B., Escorsa, E., Giereth, M., Kompatsiaris, Y., Papadopoulos, S., Pianta, E., Piella, G., Puhlmann, I., Rao, G., Rotard, M., Schoester, P., Serafini, L., and Zervaki, V. Towards content-oriented patent document processing. *World Patent Information 30*, 1 (2008), 21–33.

[34] Youtie, J., Iacopetta, M., and Graham, S. Assessing the nature of nanotechnology: can we uncover an emerging general purpose technology? *The Journal of Technology Transfer 33*, 3 (June 2008), 315–329.

Semi-automatic construction of domain-specific thesauri

Inga Kohlhof
SyynX Solutions GmbH
Nattermannallee 1
D-50829 Cologne
Inga.Kohlhof@syynx.de

Bob Schijvenaars
SyynX Solutions GmbH
Nattermannallee 1
D-50829 Cologne
schijvenaars@collexis.com

Mario Diwersy
Collexis Holdings
1201 Main Street, Suite 980
Columbia, SC 29201
diwersy@collexis.com

ABSTRACT
This paper describes a hybrid technique for the semi-automatic construction of a domain-specific thesaurus for automatic indexing. It combines statistical, linguistic and expert intelligence. To collect a list of thesaurus candidate terms we use different sources: First, we exploit existing, freely available domain glossaries. Furthermore, we use the set of keywords specified in a collection of documents that are known to be domain-specific. Eventually, using standard Natural Language Processing (NLP) techniques, we extract terms from this collection and filter out thesaurus candidate terms by applying statistical thresholds; among other things, we compare the NP's domain-specific and general frequencies. Domain experts help to determine an appropriate set of domain-specific documents and supervise the process of thesaurus construction: They check, revise and complement the resulting list of candidate terms and give the resulting list of thesaurus candidates a hierarchical structure.

Categories and Subject Descriptors
H.3.1 Thesauruses

General Terms
Design, Standardization

Keywords
term discovery, automatic term recognition, domain-specific thesaurus, semi-automatic thesaurus construction

1. INTRODUCTION
The domain-specific thesauri we construct are used for automatic indexing of documents belonging to that domain, performed by the Collexis Engine. Using the results of the indexing, the Engine calculates semantic „fingerprints“ for documents: very small, unambiguous representations of their content ([1]). These fingerprints in turn are employed for a variety of higher order purposes: for very fast document retrieval, to monitor developments and trends in fields, to calculate research profiles for single experts, expert groups, teams, institutions, departments, etc., for cities, regions etc. Examples are the Collexis DashBoard[1] and the website BiomedExperts.com[2].

For such applications we develop a process of semi-automatic thesaurus construction involving domain experts. The process consists of independent modules that can be easily exchanged if required.

Our thesauri are collections of technical terms of specific domains assigned to concept nodes. Synonyms, spelling variants and abbreviations are subsumed under a single concept node. One term is marked the preferred term for the concept in question. In addition to this equivalence relation between terms, one hierarchical relation organizes the concept nodes. The meaning of this relation is a very general one, it comprises „is-a“ relations (x is a y), part-of-relations (x is part of y) and more loose relations (x can be relevant in the context of y).[3]

Concepts can be "children" of 2 or more meta-concepts, i.e. the thesaurus is poly-hierarchical. Unlike multi-axial systems like SNOMED CT® our thesauri are mono-axial, and composition (called "post-coordination") of terms is not supported.

Further relations can be defined, definitions can be included but are not used during indexing.

By these characteristics, our thesauri are controlled vocabularies according to the classification proposed in [3] (for more information about controlled vocabularies cf. [4]).

The subtask we perform using computerized support is referred to as automatic term recognition (ATR). ATR plays a key role for domain-specific information extraction and, of course, for thesaurus construction. The history of ATR begins with the 1970s (a pioneer work is [5]). A wide variety of approaches to ATR have been developed since which can roughly be classified into linguistic, statistical or hybrid systems. A systematic survey can e.g. be found in [6], for recent developments cf. e.g. [7].

We have extracted thesaurus candidate terms for the domains of fuel cell technology, (U.S.) bankruptcy law, agriculture and the environment, and modeling and simulation in biocomputing, structural biology and cell biology.

Section 2 explains how we collect related glossaries in the web and what kinds of document collections we use as domain-

[1] http://www.collexis.com/products/knowledge-dashboard.htm

[2] http://www.biomedexperts.com

[3] The Medical Subject Headings (MeSH) thesaurus (www.nlm.nih.gov/mesh/) uses this general kind of meta-term - sub-term-relation, cf. [2]).

specific corpora. In Section 3, the steps in the process of NP extraction are described in detail. Section 4 deals with the acquisition of non-domain-specific frequencies for the extracted NPs. In 5, we specify how we use qualifying characteristics (like "being a keyword") and statistical values to arrive at a set of thesaurus candidate terms in each case. Section 6 specifies the steps necessary to make a thesaurus of a list of candidate terms. In Section 7 we give some examples for results of the actual process of term extraction. Section 8, eventually, briefly looks back and at further enhancements of the process.

The expression "term" refers to single words or groups of words that form a (linguistic) phrase.

2. SOURCES

2.1 Glossaries

When constructing a new thesaurus, our first step usually is to collect a number of glossaries freely accessible on the Internet which we use as a cross-reference (see below). We always try to find glossaries which are, according to their name and description, designed to cover the domain in question or part of it; that is: we try to rule out those that cover more than that domain to keep terms that just happen to occur in the domain-specific documents (e.g., business terms used in law documents) out of the domain thesaurus (e.g. a law thesaurus).

It may seem that the advantages of this step might depend too much on chance: there could be too few or no glossaries at all for a domain or only glossaries of limited quality. Our experiences so far, however, are encouraging: for 3 different domains, we found more than 10 glossaries, most of which were of medium-to-high quality.

We usually rank the glossaries according to their assumed quality, their size and the quality of their definitions. After normalization of the terms, we import the glossaries found worthy to be integrated.

Although glossaries are similar to controlled vocabularies they differ from thesauri as they may be designed for other purposes. Glossaries supply definitions or explanations for terms used in a certain domain which are used with a meaning different from their meaning in colloquial speech or in other domains.

By contrast, a thesaurus like the ones we make has the purpose to characterize the content of documents in a representative way, in order to make single documents and subsets of documents retrievable by factorizing sets of documents.

Thus, even terms classically present in glossaries to a domain may not occur in a domain thesaurus because no one uses them. On the other hand, thesauri may contain terms that never occur in a glossary because there is no need to define or to explain them.

2.2 Domain-specific documents

The primary sources for harvesting of candidate terms for a domain-specific thesaurus are domain-specific documents. Typically, these are sets of papers registered in bibliographic databases such as MEDLINE, the Web of Science, Scirus, etc., or abstracts of such papers. Depending on customers needs, however, other type(s) of documents will be included, e.g. reports, project proposals, books, etc.

The time range is usually limited in a natural way to digitally available documents. Ideally the documents should be representative, that is: they should cover, both in content and in content ratios, at least the important discussions in the domain in this period. The domain experts' judgment is crucial here: domain experts may directly provide the set of documents to use, formulate queries to extract the documents from databases or evaluate the suitability of a set of otherwise compiled documents.

The selection of domain-specific documents determines the maximal dimensions of the thesaurus to be constructed.

3. EXTRACTING NPs FROM DOMAIN-SPECIFIC DOCUMENTS

3.1 POS tagging

We use two parallel routes to achieve POS tagging. One is to employ two tools contained in the Natural Language Toolkit (NLTK) 0.8. NLTK is a suite of libraries and programs for symbolic and statistical natural language processing for the Python programming language.[4] Tokenization is done by the NLTK tokenizer *wordpunct*, while POS tagging is done using the Brill tagger ([9], [10]) included in the NLTK[5] trained on the Brown corpus (the latter is available at http://icame.uib.no/).

We also use our own POS tagger. This tagger was built in-house and is based on a HMM (Hidden Markov Model), combined with suffix trees.

3.2 NP extraction

As a rule, considerably more than 90% of thesaurus entries are noun phrases (NPs). While we consider extracting adjectives as candidate terms in the future, at present, we only compute thesaurus candidate terms that are NPs.

Many kinds of NPs rarely or never occur in thesauri. These include NPs starting with determiners, most kinds of complex NPs ("genetic and physical interactions", "the particular GO term with which a gene is annotated"), NPs containing deictic, relative, ordinal, quantitative, demonstrative and other elements (e.g., "your favorite molecular regulatory network", "today's healthcare", ...). The main building blocks for thesaurus terms are nouns and adjectives, occasionally further specified by prepositional phrases.

To identify the noun phrase types occurring in domain-specific documents a flat analysis as performed by a subtype of partial parsers, so-called "chunkers", is more than sufficient. Chunking (cf. in particular [11]) can be implemented with hand-crafted rules or as some machine learning technique (cf., e.g., the results of the

[4] For further information cf. http://nltk.org/, esp. [8].

[5] Copyright (C) 2001-2007 University of Pennsylvania, Authors: Christopher Maloof, Edward Loper, and Steven Bird, URL: http://nltk.sf.net

CoNLL-2000 shared task "Text chunking", described in http://www.aclweb.org/anthology-new/W/W00/W00-0726.pdf).

Since we are dealing with a limited, manageable set of NP structures and since we want full insight in and control of the parsing process at any given time, we use a rule-based chunk parser for NP detection. It is based on the chunk parser included in the NLTK toolset. The rules of our chunk parser are the tag sequences of the NPs we want to find. Technically, our chunk parser is a finite state automaton (FSA) constructed from regular expressions.

3.3 Normalization

After NP extraction the extracted NPs are normalized. NP tokens which differ only in grammatical number, letter casing (upper / lower) (incl. acronyms (all-caps-words)) or spelling are mapped to one basic form. For that purpose we apply a dictionary-based normalizer (an adapted version of the Collexis Normalizer) to the last word of each extracted NP.

3.4 Computing domain-specific and general frequencies for extracted NPs

We find candidates for a domain-specific thesaurus by comparing their frequency in domain-specific documents to their frequency in non-domain-specific documents (cf. in particular [12]). The basic assumption is: When a term is used in a set of domain-specific documents significantly more often than in a set of non-domain-specific documents, this term is more relevant for that domain than other terms.

As our corpus for non-domain-specific documents we chose a collection which covers a very broad range of topics and, at the same time, beats all other corpora by size: the web.

While more and more language scientists and technologies use the web both for corpus building and as a corpus for many different purposes (cf. [13], [14]), to use it as a "background" for comparison is a relatively new approach in ATR. Vogel reports on experiences and results of domain specific term recognition using a generic corpus as a filter, for three different domains (as part of the MITRE research project, cf. [15]). Drouin identifies corpus specific lexical items by comparing frequencies in a technical and a non-technical corpus ([16]). Asmussen describes a similar approach for enriching The Danish Dictionary (DDO) with new domain-specific vocabulary ([17]).

At present, we use the number of results to an exact search for each NP (i.e. the number of web pages that contain the exact noun phrase) in Live Search (www.live.com).

Since we only know the absolute number of occurrences, not the number of documents in the web, we use the ratio of an NP's term occurrence in the set of domain-specific documents (collection frequency, cf) and in the web (webf) (proportionally scaled, so the figures are human readable), "FreqRatio" for short:

FreqRatio = cf * 10,000 / webf.

We will refer to webf as "web frequency" below.

4. FILTERING OUT THESAURUS CANDIDATE TERMS

4.1 General approach

Since subject areas differ in size and since number and extent of the domain-specific documents considered appropriate by the experts (see 2.2) vary as well, we do not work with constant threshold values. Instead, we set and combine these values, as the case may be, in such a way that we arrive at a set of candidate terms which, according to - our - laymen's judgment, consists to a large extent (>85-90 percent) of promising candidates for a domain-specific thesaurus.

4.2 Qualifying features

4.2.1 Glossary terms

We assume that most or all of the entries of glossaries which approximately cover a domain will also be relevant thesaurus terms for that domain. Since it is not always easy to say which domain is covered by a glossary (even for an expert) we only include glossary terms that occur at least once in the domain-specific documents to ensure their relevance. The availability, number and quality of glossaries in the web varies, of course, from one domain to another (see 2.1).

4.2.2 Keywords

At least for scientific papers it is by now standard practice for authors, publishers or both to specify a small number of keywords to help prospective readers to find them (often, these must be picked from a limited list, itself a controlled vocabulary). These keywords are naturally qualified for a thesaurus constructed for this same purpose. Not only search engines and portals make use of hand-assigned keywords, but also systems for automatic indexing, term recognition and thesaurus construction (e.g., [18]). Still, as some people tend to assign keywords with very general meanings we only include those that are also used in the domain-specific documents. Unfortunately, keywords are not available for all kinds of documents (e.g. case records, project proposals).

4.2.3 Occurrence in title or abstract

Likewise, it has been common practice for a long time in ATR and thesaurus construction to assign higher weights to terms extracted from special parts of documents, especially titles and abstracts, (e.g., [19], [20]): Titles aim at putting across a realistic impression of the document's contents, the abstract is meant to present these contents in compressed form. The weight we give to these terms is dependent on the overall distribution of terms in the document set (see 4.1).

4.2.4 Word filter

We require all NPs which are not glossary terms or keywords not to contain any digits, and every word in each NP that is not a preposition to start with three letters. The primary purpose of this measure is to dispose of huge amounts of impure or false material (e.g. "“1814 /", "day$15200"). At the same time we exclude a great deal of product, type and version names, abbreviated person names and other NPs containing cardinal or ordinal numbers or quantitative specifications (e.g. ,"experiment 1", "038cu2+-sod

4", "autocorrelation coefficient of 0", ">10 mg ml x1"). In fact this filters out NPs that consist of words in the narrow sense. In doing so we accept the loss of a certain number of thesaurus candidates. In many thesauri, these kinds of NPs do not occur or occur only sporadically (e.g., "Chapter XX" is an exception in a law thesaurus). In the natural sciences and in engineering, however, code-like terms (like "3d medical image", "CB atom", "free R factor"..) occur more often, and their number raises with the "depth" of the thesaurus. If coded or abbreviated terms do not come in as a glossary term or a keyword in the first instance, they're lost. It is possible, though, to deactivate this filter in a second round (see below).

4.3 Frequency filters

4.3.1 Frequency of NPs in web pages

A certain group of NPs, namely those which are very frequent in the web, can always be excluded from the outset: They are almost certain to not contain potential candidates for any domain-specific thesaurus. A threshold of 10 million occurrences has proved a good general guide value.

The majority of NPs affected by this filter are single-word-NPs, their number is reduced by 3-15%. Among the front-runners are "result", "paper", "system", "time", "information", "page", "site". Quite regularly a group of about 20 NPs of length 2 with strongly overlapping members are also excluded, including "cell phone", "health care", "further information". Extracted NPs consisting of three or more words have not been found more than 10 million times by our web queries so far.

4.3.2 NP frequency of adjective

There is a second statistical value which enables quick exclusion of irrelevant data; it can be applied to extracted NPs with adjectives. Many adjectives which modify a noun do not convey any information about the entity denoted by that noun per se. These include deictic, relative, ordinal and quantificational modifiers ("this", "such"; "which"; "fourth", "last"; "various", "certain", "additional") and many more. For thesauri, NPs of this kind are completely irrelevant. Since they do not characterize the entities referred to by their noun, the adjectives can be combined with (almost) any noun. And therefore, the number of different nouns an adjective occurs with can be used to identify and to get rid of these NPs.

By doing so we exploit what is called the mutual informativity of noun and adjective. Since [21] mutual informativity as a measure is commonly used in NLP and ATR.

Intuitively, mutual information measures the statistical dependency of two entities (in our case of two words). Syntactically, every adjective can be combined with every NP. If the adjective and noun are statistically independent, their mutual information is zero, while if they only occur together, their mutual information is the entropy of the adjective (or noun, since their entropy is identical in this case).

An adjective's frequency/NP and its overall frequency are not, of course, statistically independent. But meaningful adjectives can be frequent. Compare, e.g., the most frequent adjectives a) by absolute number of occurrences and b) by number of different nouns it occurs with in a set of process reports.

a) such > due > legal > reasonable > new > ...

b) such > certain > new > additional > original > ...

The second and the third of the most frequent adjectives already are part of law-relevant terms (cf. "due process", "legal action").

4.3.3 Ratio of collection frequency and web frequency

To find salient terms we use the quotient of collection frequency and the web frequency as described in 3.4 ("FreqRatio"), i.e. we look for terms that are used significantly more often in domain-specific texts. The ratio of thesaurus candidates falls, the ratio of non-relevant terms rises with decreasing FreqRatio. By default we stipulate a relatively high threshold value, such that (cf. 4.3) according to laymen's assessment about 85-90% of the terms in the selected range are plausible thesaurus candidates. This means that we give more importance to precision than to recall. The lower this threshold, the larger the list of terms the expert(s) will have to pick from. As in many cases, 100% recall can only be reached by accepting a low precision. For manual review of such a list of candidate terms, a too low recall results in a too high burden for the domain experts.

By using a high threshold a certain number of domain-specific terms will always be missed: Those that, for some reason, are used about as frequently as they are used in the "background" collection from a wide range of domains. This may happen due to a number of reasons. Terms may for example be relevant for more than one domain or for a family of domains (e.g. chemicals like aldehydes are important for many domains, and so is now CO_2 emission). They could be relevant for a certain document type no matter what the content (e.g. "specific growth rate", "samples" etc. will be found in many scientific articles). They could be homonym to one or more terms with similar or different meanings (like e.g. the agricultural term "field" or the chemical term "base"), or they could be important for some domain, but at the same time be of universal relevance like, e.g. the technical terms "price" in economics or "fusion" in biology.

The frequency of NPs dramatically declines with their length. Therefore, different FreqRatio thresholds must be defined for NPs of different length. Thresholds for 2- and 3- grams can be up to 30 times higher than those for 1 grams, while it seems that NPs consisting of four words or more are so domain-specific that they are not less frequent in the net than in domain-specific texts.

Since the tagger was trained on the Brown corpus (see 3.1) which is like the net a collection of heterogeneous texts, words unknown to the tagger are from the outset more probable to be thesaurus candidates than the rest of the words. They also yield few results in web queries and therefore (usually) get lower FreqRatio thresholds.

4.3.4 Document frequency

We use three elementary frequency measures to estimate the relevance of terms in a collection of documents:

- collection frequency (cf), the total number of occurrences of a term in a set of documents
- document frequency (df), the number of documents in which a term occurs
- average document frequency (adf = cf/df)

Depending on the intended extent and depth of the target thesaurus it makes sense to require candidate terms to occur (at

least) in a certain number of documents. Document frequency can be considered a kind of sliding controller for these properties. For a thesaurus restricted to domain-specific terms relevant for most of the topics discussed in the field the minimal df value should be high. Average document frequency can be used in addition to take into account the relative importance of terms in the documents they occur in.

If a thesaurus is meant to comprehensively and precisely cover the domain, it also has to comprise terms that are used seldomly or by few authors. In this case, the required document frequency should be low, or null. In the latter case, pure collection frequency is employed to measure the relevance of terms.

It some cases it is reasonable to do a two-phase construction process. Structure and "trunk" of the thesaurus can be formed with a limited number of customary domain-specific terms in phase one to enrich it with branches and leaves in the second round.

4.4 Revision of candidate list

Before we hand it over to the experts we revise the list of candidate terms filtered out with the tools described above.

We

- restore upper/lower case spelling (cf. 3.3)
- delete expressions mistaken for NPs by the chunker due to incorrect tagging (cf. 3.1)
- delete personal names (if not personal names which are, like in "Bayes' Theorem", part of an NP)
- assign spelling variants and abbreviations to the related standard forms (cf. 3.3)
- remove semantically unqualified NPs that were not filtered out (like, e.g., "entire judgment", "different method" etc., cf. 4.3.2).

Time spent on the manual revision depended mainly on the number of terms; we expended between 4 and 14 person hours (cf. table 1).

4.5 Thesaurus candidate lists and existing thesauri

For a number of reasons, it can be useful or desirable to include existing thesauri in the process of generating a candidate list. For instance, the candidate list may be meant to extend an existing thesaurus in which case no term of the existing thesaurus should occur in the candidate list. Or an existing thesaurus can serve as a cross-reference filter for the candidate list to be generated. In that case, the terms of the existing thesaurus will be treated like glossary terms (cf. 4.2), i.e. when they occur in the domain-specific documents, they will be included in the thesaurus candidate list.

The thesauri factored in may be public domain, or discretionary – like MeSH – or proprietary thesauri.

For example, when we built up the thesaurus candidate list for the domain of modeling and simulation in biocomputing, structural biology and cell biology we included about 6,000 terms from the Research, Condition, and Disease Categorization (RCDC) thesaurus[6] and excluded 3,500 MeSH terms.

5. TURNING THE CANDIDATE LIST INTO A THESAURUS

5.1 Selecting thesaurus terms from the candidate list

In the next step, domain experts pick the terms for the domain-specific thesaurus from the list of candidates. As yet conclusive results are present to us for three cases (cf. Table 1 in the Appendix). From our first candidate list (compiled 2/2008) the fuel cell expert picked 775 or 37% out of 2101 candidate terms as thesaurus terms. At the same time, he added 23 terms. The Bankruptcy law experts accepted 1,532 or 91% of 1,689 terms proposed in October 2008, and added 36. Eventually the specialists for modeling and simulation in biocomputing, structural biology and cell biology decided to accept the complete list of candidates as thesaurus terms (12/2008). We still have not gotten any feedback from the agriculture/environment group.

These figures also reveal a learning process: After our experiences with the first thesaurus we decided to spend more time revising our candidate lists.

When experts decide to accept, to reject and to add terms they evaluate our process of term recognition - to date this is the only evaluation taking place. It is deficient in a number of ways. Among other things it is usually performed by only one person, no explanation for the choice of terms is documented, experts may feel unable to judge the relevance of proposed terms or lack the time to think about missing terms etc. Still, this kind of evaluation is a very direct one - the experts modify the candidate list immediately before it is put in use.

5.2 Structuring the thesaurus

To enable quick and comfortable searching after automatic indexing by the Collexis Engine, thesaurus terms are classified in about 10 to 15 generic categories. Depending on the size of the thesaurus it can make sense to add another hierarchical level, that is to further subclassify the terms with intermediate generic terms. Like with the selection of domain-specific documents domain experts may structure the terms in content or check, correct and complement our proposals.

6. RESULTS

This section gives some examples for the effects of the cooperation of the described filters. We will refer to the thesaurus candidates for the domains of fuel cell technology, (U.S.) bankruptcy law, agriculture and environment, and modeling and simulation in biocomputing, structural biology and cell biology as "fuelcells", "law", "agric" and "biocomp", respectively.

[6] further information at rcdc.nih.gov/category_process/

Table 1 (Tables and diagrams are to be found in the Appendix) shows some skeleton data and gives an impression of the dimensions of the NP filtering process for the four thesauri. About 5% of the different NPs found in the documents are selected as thesaurus candidates.

Using the bankruptcy law thesaurus as an example, tables 2 and 3 illustrate the use and effects of some filters. Table 2 shows how we first applied a set of filters to all extracted NPs and how they reduced the number of NPs. After that first reduction we defined more specific filters for the remaining NPs of different length. Table 3 shows the specific filters used for the filtering of law thesaurus candidate terms and their effects.

Diagram 1 gives the numbers of candidate terms selected for the 4 thesauri, broken down by their length. More than 80% of the candidates are 1- and 2-grams. In most cases single words outnumber the 2-grams, while 5- and 6 grams are very unlikely to be selected. All this corresponds to the structure of the glossaries we used (cf. 2.1).

7. CONCLUSIONS

The reported experiences with our semi-automatic, modular process for thesaurus development are encouraging. We believe that with our hybrid method - building on existing accessible knowledge (the glossaries), using text-structure along with linguistic and statistical information, and working with domain experts - we are on the right track to construct functional thesauri for automatic indexing. The experts will also help us to improve the process according to users' needs and preferences. We will soon include a module to extract terms co-occurring with "seed" terms[7]. In the medium term we want to automatically support synonym detection and thesaurus structuring.

8. REFERENCES

[1] van Mulligen Erik M., Diwersy, Mario, Schmidt, Martin, Buurman, Henk, and Mons, Barend 2000: Facilitating networks of information. Proc AMIA Symp. 2000:868-72.

[2] Basili, R., Vindigni, M. and Zanzotto, F. M. (2003): Integrating ontological and linguistic knowledge for conceptual information extraction. In: Proceedings of the IEEE/WIC WI-2003, Conference on Web Intelligence, Halifax, CA.

[3] Ingenerf, Josef 2007. Die Referenzterminologie SNOMED CT®. Von theoretischen Betrachtungen bis zur praktischen Implementierung. Lübeck: mmi.

[4] ANSI/NISO Z39.19-2005 - Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies. Available at www.niso.org/standards/resources/Z39-19-2005.pdf.

[5] Sparck-Jones, Karen 1970. Automatic thesaurus construction and the relation of a thesaurus to indexing terms. In: Aslib Proceedings 22 (5), pp. 226-233.

[6] Kageura, Kyo and Umino, Bin 1996. Methods of automatic term recognition: A review. Terminology 3:2, 259 ff.

[7] Collexis Research Profiles and Mediator already benefit from co-occurring terms.

[7] Zhang, Ziqi, Iria, José, Brewster, Christopher and Ciravegna, Fabio 2008. A Comparative Evaluation of Term Recognition Algorithms. Available at www.dcs.shef.ac.uk/~kiffer/papers/Zhang_LREC08.pdf.

[8] Bird, Steven and Loper, Edward 2004. NLTK: The Natural Language Toolkit. In: Proceedings of the ACL demonstration session. Barcelona: Association for Computational Linguistics, pp. 214-217.

[9] Brill, Eric 1992. A simple rule-based part of speech tagger. Proceedings of the Third Annual Conference on Applied Natural Language Processing, ACL.

[10] Brill, Eric 1995. Unsupervised learning of disambiguation rules for part of speech tagging.

[11] Abney, Steven 1991. Parsing By Chunks. In: Robert Berwick, Steven Abney and Carol Tenny (eds.), Principle-Based Parsing. Kluwer Academic Publishers, Dordrecht.

[12] Ahmad, Khurshid, Davies, Andrea, Fulford, Heather and Rogers, Margaret 1992. What is a term? The Semi-Automatic Extraction of Terms from Text. In: Mary Snell-Hornby et al. (eds.) Translation Studies - an Interdiscipline. Amsterdam / Philadelphia: John Benjamins Publishing Company.

[13] Kilgarriff, Adam and Greffenstette, Gregory 2003. Introduction to the. Special Issue on Web as Corpus. In: Computational Linguistics, Volume 29, No 3, September 2003: Special Issue on the Web as Corpus, pp. 333-347.

[14] Hundt, Marianne, Nesselhauf, Nadja and Biewer, Carolin (eds). (2007). Corpus Linguistics and the Web. Amsterdam/New York: Rodopi.

[15] Vogel, David 2003. Using generic corpora to learn domain-specific terminology. In: Steve Donoho, Tom Dybala, Marko Grobelnik, Natasa Milic-Frayling, and Dunjy Mladenic (eds.): Proceedings of the ACM SIGKDD '02 Workshop on Link Analysis for Detecting Complex Behaviour (LinkKDD '03), Washington, DC, USA, August 27, 2003.

[16] Drouin, Patrick. 2003. “Term extraction using non-technical corpora as a point of leverage”. Terminology 9:1, 99–115.

[17] Asmussen, Jørg 2005. Automatic detection of new domain-specific words, using document classification and frequency profiling. In: Proceedings of the Corpus Linguistics 2005 conference, Vol 1., Birmingham.

[18] Utiyama, Masao, Murata, Masaki and Isahara, Hitoshi 2001. Using author keywords for automatic term recognition. In: Terminology, Volume 6, Number 2, pp. 313-326.

[19] Harman, Donna K. (ed.). The First Text REtrieval Conference (TREC-1), NIST SP 500-207, National Institute of Standards and Technology, Gaithersburg, MD, 1993.

[20] Trotman, Andrew 2005. Choosing document structure weights. In: Information Processing & Management, Volume 41, Issue 2, March 2005, Pages 243-264.

[21] Church, K. W., Hanks, P. (1990). "Word Association Norms, Mutual Information, and Lexicography". In: Computational Linguistics, Vol. 16, No. 1, pp. 22-29.

[22] The Brown Corpus, available at http://icame.uib.no/.

Appendix A. Tables and Diagrams

Table 1. Skeleton data of NP filtering for different thesauri

	fuel cells	law	agric	biocomp
document type	abstracts	process reports	abstracts	papers
number of documents	2,660	6,000	33,368	123
words therein	298,648	18,135,695	5,029,555	1,449,269
extracted NPs	101,958	10,251,839	3,384,286	804,990
normalized extracted NPs	42,235	937,216	724,338	207,769
NPs selected	2,435	9,664	1.096	2,187
glossary terms extracted	-	313	-	437
keywords extracted	2,428	-	-	131
candidate NPs (after manual revision, cf. 4.4)	2,101	1,689	828	1,971
person hours expended for manual revision	4	8	5	14
NPs accepted by experts (% compared to ^)	775 (37%)	1,532 (91%)	yet unknown	1,971 (100%)

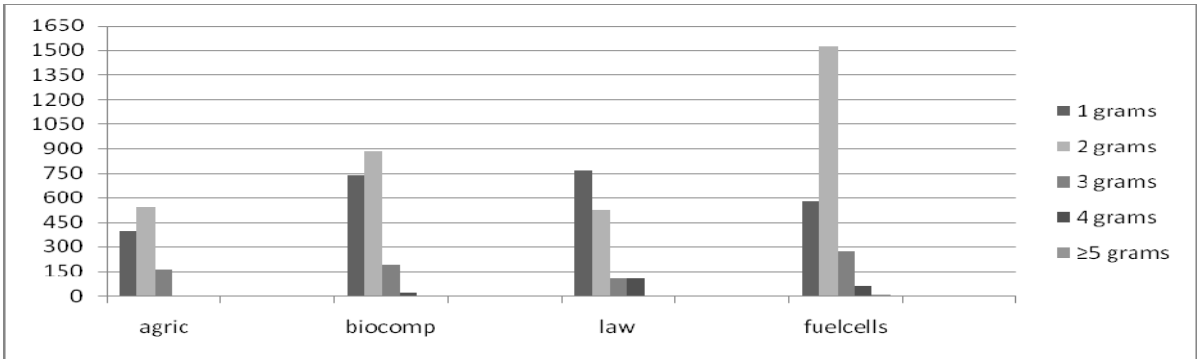
Table 2. Bankruptcy Law Thesaurus Candidates: Filters applied to all extracted NPs, and their effects

Filters applied to all extracted NPs	value	number of NPs	relative # NPs (%)
Normalized extracted NPs		937,216	100%
Word Filter		116,616	12,4%
Document Frequency	> 2	140,753	15,0%
Max. no. of different NPs occ. w. adjective	< 2303	114,191	12,2%
Global occ. of adjective	< 22,113	115,707	12,3%
Web frequency	< 10 Mio	112,899	12,1%
All filters applied		110,474	11,8%

Table 3. Bankruptcy Law Thesaurus Candidates: Filters applied to NPs of different length, and their effects

No of terms after appl. of global filters (cf. table 2)	**1 grams** (unknown)		**1 grams** (known)		**2 grams**		**3 grams**		**≥4 grams**	
110,474		3,634		18,408		47,969		22,300		18,16
Filters for N gram classes and their effects										
Coll. freq.	> 2	1,063	>2	18,353	>10	14,746	> 5	5,335	> 10	3,891
Prop. Doc. Freq.	≥ 1%	973	≥ 1%	1,681	≥ 0,23%	7,982	≥ 0,1%	3,249	≥ 0,3%	1,669
Web-Freq	> 0	3,364	> 0	17,109	< 5	44,420	> 0	6,435	>0	14,03
FreqRatio	≥ 0,01	2,830	≥ 0,02	17,713	> 10	26,852	> 10	7,829	>500	6,478
NPs filtered out		**415**		**1,664**		**4,610**		**2,373**		**611**

Diagram 1. Absolute numbers of candidate terms filtered out for the 4 thesauri, broken down by length



Web Directories as Topical Context

Rianne Kaptein[1] Jaap Kamps[1,2]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] ISLA, Informatics Institute, University of Amsterdam

ABSTRACT

In this paper we explore whether the Open Directory (or DMOZ) can be used to classify queries into topical categories on different levels and whether we can use this topical context to improve retrieval performance. We have set up a user study to let test persons explicitly classify queries into topical categories. Categories are either chosen freely from DMOZ, or from a list of suggestions created by several automatic topic categorization techniques. The results of this user study show that DMOZ categories are suitable for topic categorization. Either free search or evaluation of a list of suggestions can be used to elicit the topical context. Free search leads to more specific topic categories than the list of suggestions. Different test persons show moderate agreement between their individual judgments, but broadly agree on the initial levels of the chosen categories. When we use the topic categories selected by the free search as topical context, this leads to significant improvements over the baseline retrieval results. The more general topic categories selected from the suggestions list, and top level categories do not lead to significant improvements.

1. INTRODUCTION

One of the main bottle-necks in providing more effective information access is the poverty of the query end. With an average query length of about two terms, users provide only a highly ambiguous statement of the, often complex, underlying information need. This significantly restricts the ability of search engines to retrieve exactly those documents that are most relevant for the user's needs. Associating the query with a topical category can help to disambiguate the query. If query topics can successfully be associated with topic categories, this topical context can be used in different ways i.e. to improve retrieval effectiveness, to filter out results on non-relevant topic categories or to cluster search results. In this paper we will investigate how to get and use topical context on different levels of granularity. Queries can be associated with a topical category by using implicit or explicit techniques. Examples of identifying topical context implicitly are using a user profile built on previous information seeking behavior, previous issued queries, or automatic classification of query words or retrieved documents. We will elicit the context explicitly, i.e. ask the user to classify a query into a topical category.

Several large directories on the web have organised their information into topical categories, usually in a hierarchical way e.g. DMOZ (also known as ODP Open Directory Project) [5], Yahoo! Directory [18] and Wikipedia [17]. There has been a stream of papers that use some form of topical model or context use the DMOZ directory, or a part of it, to represent topical categories (see Section 2 below). DMOZ has a lot of attractive features, it is hierarchical, large, and created by human users especially for the web. In a previous study [11] we have used a small number of self-defined categories, that did not cover a wide range of query topics. By using a considerable part of the DMOZ category we can cover a wide range of topics. For these reasons this paper uses the DMOZ directory to represent topical categories.

Being large also has some disadvantages, for users it might not be so easy find the category they are searching for. There is a trade-off between the quality of the user categorization, i.e. whether the category covers exactly the query topic, and the effort that is needed. Searching or browsing the complete directory requires the most effort from the user, but can result in finding categories an automatic classifier can not find. Choosing from a list of suggestions takes less effort from the user, but there is always a risk that the best possible topic category is not included in the list of suggestions. We will examine whether there is also a trade-off between the level of categorization, and retrieval effectiveness when the topical context is used. We expect that low level and thus specific categories will prove most beneficial for retrieval effectiveness.

In this paper we address the following main research question:

- Can we effectively use the DMOZ directory as a source of topical context?

We break up our main research question, into the following two research questions:

1. Can the DMOZ directory be used to effectively categorize query topics into topic categories?

We carry out a user study that identifies topical context explicitly in order to answer our first research question. We explore whether the topic categories in DMOZ are representative for query topics. Furthermore, we compare two different forms of deriving context explicitly, i.e. free search or browsing on the DMOZ site, and evaluation of topic categories from a list of suggestions. Agreement on the relevance of DMOZ categories between different test persons is also considered. To answer our second research question, we use the results from our user study to look at the effects of using topical context on retrieval performance:

2. Can we use topical context to improve retrieval effectiveness?

We compare performance of runs using topical context in addition to the query on different levels in the DMOZ directory.

The rest of this paper is organized as follows. In the next section we discuss related work. In Section 3 we describe the data, i.e. the

DIR 2009, February 2-3, 2009, Enschede, the Netherlands.

queries that we use and the DMOZ directory. Section 4 describes the language models that we are using for topic categorization and retrieval. In Section 5 we discuss the user study we have executed. Section 6 describes experiments where we use the topical context that we got from our user study to try to improve retrieval effectiveness. Finally in Section 7 we discuss the results and draw our conclusions.

2. RELATED WORK

There is a range of studies that use topical models to improve retrieval performance or retrieval effectiveness [1, 2, 4, 6]. Two approaches are commonly used, one approach creates some kind of user profile that does not depend on the query. These user profiles can be built in different ways. Chirita et al. [4] lets users pick multiple topic categories from DMOZ to create user profiles which best fit their interests. At run-time the output of the search engine is reranked by using a calculated distance from the user profile to each output URL.

Liu et al. [12] builds user profiles automatically by using the search history, that consists of the issued queries, relevant documents and related categories. A new query is mapped to a set of categories using the user profile, a general profile, or a combination of user and general profile. The categories are ranked, and the top 3 categories are chosen to reflect the user's search intention.

Also Trajkova and Gauch [14] builds user profiles based on the user's search history. Web pages that a user has visited for at least a minimum amount of time are classified into a category from the DMOZ directory. Only the top 3 levels of the directory are used. To classify a Web page, the highest weighted 20 words are used to represent the content of the Web page. Classification consists of comparing the vector created for the Web page with each category vector (created and stored during training) using the cosine similarity measure.

The other approach, which is also employed in this paper, is to use topical models that depend on the query. Wei and Croft [16] manually assign topic categories according to some basic rules. Haveliwala [6] considers two scenarios to assign topical categories to queries. Both scenarios use personalization vectors calculated for the 16 top-level DMOZ categories. In the first scenario, unigram language models are used to calculate the class probabilities given a query for each of the 16 top-level DMOZ categories. The three categories with the highest probabilities, are selected to compute topic-sensitive PageRank scores. In the second scenario context of the query is taken into account. For example, users can highlight a term in a Web page, and invoke a search. The context, in this case the Web page, is then used to determine the topic. Instead of only the query terms, the terms of the whole page are used to rank the 16 top-level DMOZ categories. Two other sources of query context are also suggested. First, using the history of queries issued leading up to the current query. Second, if the user is browsing some sort of hierarchical directory, the current node in the directory that the user is browsing at can be used as context. Potential query independent sources of context include the users' browsing patterns, bookmarks, and e-mail archives.

Bai et al. [2] compares the automatic and the manual assignment of topical domains. Here, the topic domains do not come from an existing topic hierarchy, but the users can define their own domains. Domain models are created by either using the relevant documents for the in-domain queries, or by using the top 100 documents retrieved with the in-domain queries. TREC queries 51-150 are used, since these query topics also include a manually assigned topic domain. Automatic query classification is done by calculating KL-divergence scores. Although the accuracy of the automatic query classification is low, the effectiveness of retrieval is only slightly lower than when the query domain is assigned manually.

Besides topical context, a well-studied form of context is genres of webpages. For example, Rosso [13] explores user-based identification of web genres. He defines genre as: essentially a document type based on purpose, form, and context. Examples of genres are resumes, scientific articles or tax income forms. The study contains of three parts, first information is obtained on what genres users perceive. Secondly, all used terminology is refined into a tentative genre palette. Finally, the genre palette is validated by letting users classify pages into the defined genres.. The study is restricted to pages from the edu domain to increase the chance of developing a recognizable palette.

3. DATA

In this paper we investigate whether we can use the DMOZ directory as a source of topical context. We use ad hoc topics from the TREC Terabyte tracks as test data. The TREC Terabyte track ran for three years, and provides us with 150 ad hoc topics that consist of three components, i.e. title, description and narrative. The title field contains a keyword query, similar to a query that might be entered into a web search engine. The description is a complete sentence or question describing the topic. The narrative gives a paragraph information about which documents are considered relevant and/or irrelevant. All topics are created by NIST assessors [3].

The web collection that is used to search relevant pages for these topics is the .GOV2 collection, a collection of Web data crawled from Web sites in the .gov domain during early 2004. Topics are only created if the .GOV2 collection contains relevant pages for the topic. The DMOZ directory is intended to cover the whole Web, thereby also including the .gov domain. In total, around 1% of the sites listed in the DMOZ directory is from the .gov domain. Some of the DMOZ categories hardly contain any sites from the .gov domain, e.g. games, shopping and sports. The categories health, regional and science contain the most sites from the .gov domain. We expect therefore that also most topics will be categorized into the categories health, regional and science.

The DMOZ directory is organized as a tree, where the topic categories are inner nodes and pages are leaf nodes. Nodes cannot only have multiple child nodes, but by using symbolic links, nodes can appear to have several parent nodes as well. Since the DMOZ directory is free and open, everybody can contribute or re-use the dataset, which is available in RDF. Google for example uses DMOZ as basis for its Google Directory service [4].

The complete DMOZ directory contains over 590,000 categories. Categories selected by test persons during the free search can be any of the 590,000 categories, except categories under the "World" category, that contains categories in languages other than English. It does not matter if the category contains links to webpages or not. We allow multiple DMOZ categories to be assigned to one topic.

To produce the list of suggestions, we focus on a part of the DMOZ directory in order to reduce complexity. That is, we use mainly categories from the first four levels of DMOZ, which still comprises around 30,000 categories. In addition we consider a classification on the top level of the DMOZ directory, which comprises of 15 topic categories.

4. MODELS

Topical context can be derived either implicitly or explicitly. In this paper we focus on explicitly derived topical context that is obtained from a user study. We first describe the language modeling

approach, followed by the models for topic categorization that are used to generate a list of suggested categories. These same models could be used to derive topical context implicitly. In the last part of this section, we describe the model we use to incorporate topical context in our retrieval model.

4.1 Language Modeling

We use unigram language models [7] for topic categorization as well as for retrieval. Our standard model for document retrieval uses Jelinek-Mercer smoothing [19] in a mixture of the document model with a general collection model as follows, i.e., for a collection C, document D, query Q and smoothing parameter λ:

$$P(Q|D) = \prod_{t \in Q} \left((1-\lambda)P(t|D) + \lambda P(t|C)\right),$$

where

$$\begin{aligned} P_{mle}(t|D) &= \frac{tf_{t,D}}{\sum_t tf_{t,D}} \\ P_{mle}(t|C) &= \frac{\text{doc\_freq}(t,C)}{\sum_{t' \in C} \text{doc\_freq}(t',C)} \end{aligned}$$

Instead of using maximum likelihood estimation to estimate the probability $P(t|D)$, it can also be estimated using parsimonious estimation. The parsimonious model concentrates the probability mass on fewer terms than a standard language model. Terms that are better explained by the general language model $P(t|C)$ (i.e. terms that occur about as frequent in the document as in the whole collection) can be assigned zero probability, thereby making the parsimonious language model smaller than a standard language model. The model automatically removes stopwords, and words that are mentioned occasionally in the document [8].

The model is estimated using *Expectation-Maximization*:

$$\begin{aligned} \text{E-step:} \quad & e_t = tf_{t,D} \cdot \frac{\alpha P(t|D)}{\alpha P(t|D) + (1-\alpha)P(t|C)} \\ \text{M-step:} \quad & P(t|D) = \frac{e_t}{\sum_t e_t}, \text{i.e. normalize the model} \end{aligned}$$

In the initial E-step, the maximum likelihood estimates are used to estimate $P(t|D)$. The E-step benefits terms that occur relatively more frequent in the document as in the whole collection. The M-step normalizes the probabilities. After the M-step terms that receive a probability below a certain threshold are removed from the model. In the next iteration the probabilities of the remaining terms are again normalized. The iteration process stops after a fixed number of iterations or when the probability distribution does not change significantly anymore. For $\alpha = 1$, and a threshold of 0, the algorithm produces the maximum likelihood estimate $P_{mle}(t|D)$ as defined before. Lower values of α result in a more parsimonious model. We will denote the resulting estimate by $P_{pars}(t|D)$.

4.2 Topic Categorization

We will discuss three methods to automatically categorize topics. The first two methods are similar, and consist of two steps. In the first step we create topical models of the DMOZ topic categories. Secondly, we assign a topical category to each query by using either the query title, or the top 10 retrieved documents.

To create a topical model for a topic category, we crawl the sites from the category, and of all its available direct sub categories. If that results in at least 10 sites a parsimonious language model is created using the Expectation-Maximization equation described in the previous section. Instead of document D we now have TM, the topical model, that consists of the raw text of the web sites belonging to the category. The background collection C here is the DMOZ background corpus. It consists of the raw text of all web pages up to level 4 we were able to crawl. All terms with term frequency 1 are excluded from the background corpus. The corpus consists of 459,907 documents and a total number of 350,041,078 terms.

The websites used to create the topical model are spread over the category and all its subcategories. For efficiency reasons we have crawled only the upper four levels of the DMOZ directory, so we can create topical models up until the third level of the hierarchy using also the subcategories. The topical models on the fourth level use only the sites on that level.

The second step is to assign a topical category to each query. Our first method is based on classifying documents.

Top ranking documents We use the top 10 results of a baseline model run, and select categories fitting these documents best.

The documents are classified into a topical category as follows. First, the documents are scored on DMOZ top level categories by scoring each of the top level topical models on the documents:

$$P(TM|D_{top}) = \sum_{d \in D_{top}} \prod_{t \in d} \left((1-\lambda)P(t|TM) + \lambda P(t|C)\right),$$

The topical models ranked by their probabilities are saved. The documents are then classified into the second-level categories. Similarly, the documents are classified into the third and fourth level categories, but for computational efficiency here only sub categories from the 20 highest ranked topic categories are used. In the end, the topical category belonging to the topical model with the highest probability, no matter on what level, is assigned to the query.

Our second method is directly classifying the query.

Query We simply classify the short topic statement in the title field, and select best matching categories

In this case, the top level topical models are scored on the query.

$$P(TM|Q) = \prod_{t \in Q} \left((1-\lambda)P(t|TM) + \lambda P(t|C)\right),$$

Again the topical models are ranked by their probabilities, and the process continues in the same way as the top 10 result classification.

The third method we use to categorize the query is simple.

Title match We match the query words with the label of the topic category.

If all query words are present in the topic category label, the topic category is assigned to the query. When a topic category matches all query words, all its descendants automatically also match all query words. However, we then only assign the highest level topic category. Both the query words and the topic category labels are stemmed using a Porter stemmer. This method only assigns a topic category to a query topic if there is an exact match.

To produce a list of suggestions for a topic, we merge the top 10 ranked categories from the three categorization methods

4.3 Retrieval

For retrieval we use not only the query, but also a topical model assigned to the query topic. To produce a ranking a mixture of the query model and the topical model is used as follows:

$$P(Q, TM|D) = (1-\beta)(P(Q|D) + \beta(P(TM|D))$$

Table 1: Coverage of topics: taking all evaluations, and taking the best evaluation per topic.

	Not relevant		Too broad		Too specific		Excellent	
	All Evals	Best Eval	All Evals	Best Eval	All Evals	Best Eval	All Evals	Best Eval
Suggested:								
Query	78.7% (1,193)	14.1% (19)	15.8% (239)	45.2% (61)	3.6% (54)	15.6% (21)	2.0% (30)	25.2% (34)
Top Docs	77.2% (1,188)	11.1% (15)	19.8% (304)	60.7% (82)	1.9% (29)	15.6% (21)	1.1% (17)	12.6% (17)
Rel Pages	79.4% (1,212)	4.0% (2)	18.1% (276)	54.0% (27)	1.6% (25)	18.0% (9)	0.9% (13)	24.0% (12)
Title Match	17.9% (5)	0.0% (0)	17.9% (5)	0.0% (0)	21.4% (6)	14.3% (2)	42.9% (12)	85.7% (12)
Total	80.1% (2,861)	1.5% (2)	15.8% (563)	45.2 % (61)	2.6% (93)	17.8% (24)	1.6% (56)	35.6% (48)
Free Search:								
First Cat.	3.4% (8)	1.5% (2)	14.8% (35)	9.0% (12)	43.5% (103)	35.3% (47)	38.4% (91)	54.1% (72)
Second Cat.	5.2% (3)	4.3% (2)	22.4% (13)	13.0% (6)	56.9% (33)	63.0% (29)	15.5% (9)	19.6% (9)
Total	3.7% (11)	1.5% (2)	16.3% (48)	9.0% (12)	46.1% (136)	35.3% (47)	33.9% (100)	54.1% (72)

$P(TM|D)$ is estimated similarly to $P(Q|D)$ as described before.

$$P(TM|D) = \prod_{t \in TM} \left((1-\lambda)P(t|D) + \lambda P(t|C)\right),$$

5. USER STUDY

In this section we describe the user study that has been executed in order to let test persons assign topic categories to query topics.

5.1 Design

The user study is designed as follows. Test persons first read an instruction, and do a training task. Before starting the actual tasks, test persons fill out a pre-experiment questionnaire that consists of some demographic questions. The main part consists of 15 tasks. Each task corresponds to one topic. At the beginning of each task the topic, consisting of query title, description and narrative, is given. Each task is then divided into four subtasks:

1. Pre-task questions
2. The evaluation of a list of suggested categories.
3. Search or browse on the DMOZ site to find the best category.
4. Post-task questions

In the second and third task also some questions are asked on how easy the task was, and how confident the test persons are about their categorization. After the 15 tasks each test person fills out a post-experiment questionnaire that consists of questions on how they experienced and liked the different tasks. At each stage of the user study, there are open questions for comments of any kind.

In subtask 2 the test person evaluates a list of suggested categories. For each suggestion the test person evaluates how relevant the category is to the topic by answering the question: "For each suggested category evaluate how relevant it is to the topic". The four options are: "Not at all", "Relevant, but too broad", "Relevant, but too specific", and "Excellent". The list of suggestions is composed of the categories resulting from the three topic categorization methods described in the previous subsection.

In subtask 3 the test person is free to select a category from the DMOZ site that he or she thinks applies best to the topic. Categories can be found by browsing the DMOZ site, or by using the search function on the DMOZ site. If the test person finds more than one category that applies best to the query topic, there is a possibility to add a second DMOZ category. The test person evaluates again the relevance of the found category to the topic. We do not rotate subtask 2 and 3 because our goal is to obtain good human feedback. Seeing the list of suggestions first means there is a learning effect which can improve the quality of the categories selected in the free search.

5.2 Set-up

The user study is done using the queries from the three TREC Terabyte tracks 2004, 2005 and 2006 (.GOV2 collection of 25M documents) [15]. Topics 801-850 are done by two to four test persons, all other topics are done by one test person. In total 135 out of the 150 Terabyte topics are covered. The order and the selection of topics is randomized. Each test persons gets assigned 15 topics.

For the automatic query topic categorization we have to set some parameters. We use the topic categorization methods as described in Section 4.2, where $P(t|TM)$ is calculated according to the parsimonious model, $P_{pars}(t|TM)$. Stopwords are removed according to a standard stopword list. Stemming is not applied.

The standard value of the smoothing parameter λ in the language model is 0.15. In the TREC Terabyte tracks, it is known that the .GOV2 collection requires little smoothing [9], i.e. a value of 0.9 for λ gives the best results.

For the parsimonious model we have to set the parameters α and the threshold parameter. We set the threshold parameter at 0.0001, i.e. words that occur with a probability less than 0.0001 are removed from the index. We set $\alpha = 0.1$ for the parsimonious model, based on initial experiments with a part of the topic set.

The online user study records all answers, and also the time it takes test persons to do the different tasks. The open text answers, i.e. copying the URL from the DMOZ site, are manually preprocessed before the analysis to ensure they are all in the same format.

5.3 Results

In this section we discuss and analyze the results of the user study.

Demographics

The user study has been filled out by 14 test persons, of which 9 male and 5 female. Two test persons participated twice in the user study, so they did 30 instead of 15 topics. The main part of the test persons is studying or working within the field of information retrieval. Average age is 31 years. Half of them are familiar with the DMOZ directory, and 3/4 of them are familiar with the subject of topic categorization.

Appropriateness of DMOZ categories

We first look at the question: does an appropriate DMOZ category exists for the topics? In Table 1 we present the coverage of the query topics, that we get from the answers to the question of how relevant the suggested and the free search categories are to the top-

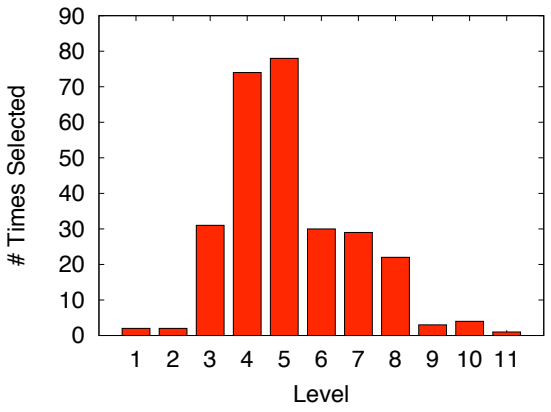


Figure 1: Levels of DMOZ categories selected by free search

Table 2: Free search vs. Suggestions list results

	Free Search		Suggestions	
	Avg.	Final	Avg.	Final
Time in min.	2.0		1.3	
Quick		3.5		3.5
Confident	3.5	3.4	3.5	3.4
Easy	3.0	3.2	3.2	3.5

ics. The columns with 'All Evals' are all evaluations per topic of all test persons taken together. The 'Best Eval' columns take only the best evaluation of all test persons and categories per topic, where a 'Too specific' category is rated above 'Too broad'. To produce the suggested categories a fourth categorization method was included. Similar to the 10 top ranked documents, 10 randomly chosen relevant documents are used for categorization (Rel Pages). We can see that when the list of suggestions is used, for only 1.5% of the topics no relevant DMOZ category is found. When the category is relevant, it is usually too broad (45.2% of the topics). Still, for 35.6% of the topics and excellent matching category is found. When free search is used, also for 1.5% of the topics no relevant category is found. For more than half of the topics (54.1%) an excellent matching category is found.

Next, we look at the question: what is the level in the DMOZ hierarchy where the most suitable DMOZ categories reside? With free search the test persons can select a category on any level of the DMOZ directory. Figure 1 shows the distribution of categories over the level of the DMOZ hierarchy. We see that the deepest level that is chosen is 11. The median level is 5.

List Selection versus Free Search

We now turn to the two ways of eliciting explicit category feedback: either by selecting from a list of suggestions, or by freely searching the DMOZ hierarchy.

Table 2 compares free search with the evaluation of the suggestions on different variables. Variables 'Quick' (I directly found the selected category(ies), and did not browse in several categories), 'Confident' (I am confident I selected the best possible category(ies)) and 'Easy' (It was easy to select categories) are measured on a Likert-scale from 1 to 5, where 1 means 'Strongly Disagree' and 5 means 'Strongly Agree'. Average numbers are averaged over all test persons and all topics. The final numbers are averages over all test persons on answers in the post-experiment questionnaire. When comparing the free search with the evaluation of suggested categories, we have to consider a bias that occurs because the test persons always first evaluate the list of suggested categories and then do the free search. In close to 50% of the cases, the test persons say the list of suggestions helped them to select a category from the DMOZ site using free search. In 55% of the cases the test persons think that the category they selected freely from the DMOZ site is better than all the suggestions in the list.

How easy and how efficient are both methods of eliciting explicit topical context? The average time spent per topic for the free search is higher than the average time spent for the evaluation of the suggested categories (2.0 minutes and 1.3 minutes respectively). The test persons however perceive both methods to be as quick. The confidence in their classifications is the same on average, and in the final evaluation for both methods. The test persons find the evaluation of the suggestions list slightly easier than the free search.

When asked what method the test persons prefer, the replies are mixed. 3 test persons prefer free search, 4 test persons prefer evaluation of a suggestions list, and 7 test persons prefer to look at a list of suggestions, and then search freely on the DMOZ site.

Agreement between Test Persons

We now look at the agreement between different test persons categorizing the same topic. We calculate pairwise agreement between test persons. Strict agreement means there is agreement on the relevant categories, and on the degree of relevance ('Relevant, but too broad', 'Relevant, but too specific', and 'Excellent'). Lenient agreement means there is agreement on the relevant categories, but the degree of relevance is not taken into account. Categories that are evaluated as not relevant by all test persons are not included.

For the suggestions list two types of agreements are calculated. 'All evaluations' calculates agreement for each category on the suggestions list when at least one test person considers the category relevant. One combination of different methods is used on the suggestions list, i.e. a category is only selected if both the classification of top 10 retrieved documents and the query produce the category (see Combination in Table 3). 'Best match' only calculates agreement for the category of the suggestions list with the best agreement. Similarly, when free search is used, and two topic categories are selected, only the best matching categories are used to calculate agreement. Agreement is calculated on different levels, where categories are simply cut off at the desired level. The 'Complete' row gives agreement on the complete topic categories without cut off. The results are presented in Table 3.

What is the agreement between test persons? Strict agreement for the suggestions list total and the free search is almost the same, 0.14 and 0.15 respectively. Categories selected by free search receive somewhat higher lenient agreement than the categories from the list of suggestion, 0.20 and 0.34 respectively.

What is the difference in agreement over the different list suggestion methods? From the three methods used to produce categories for the list of suggestions, the query title match produces the categories that best cover the query topic, and that receive the most agreement. The drawback of this method, is that only for a small percentage of topics, there is an exact match with a DMOZ category label (6). Expanding this method to include nearly exact matches could be beneficial. The combination of methods also achieves better agreement than the separate methods, on a larger number of topics (23).

Every chosen category in the DMOZ hierarchy is subcategory of a whole path up to the root node. So different categories may still share the same top-level categories. What is the agreement over levels of the DMOZ hierarchy? We look here at the best matching

Table 3: Strict and lenient agreement between test persons over all relevant judgments, and over best matching relevant judgements.

	# topics	Strict	Lenient
List (All evaluations)			
Query	44	0.12	0.22
Top Docs	49	0.14	0.18
Rel Pages	48	0.15	0.18
Combination	23	0.28	0.38
Title Match	6	0.69	0.89
Total	50	0.14	0.20
List (Best match)			
Level 1	50	–	0.75
Level 2	50	–	0.73
Level 3	48	–	0.67
Level 4	37	–	0.48
Complete	50	0.61	0.75
Free Search (Best match)			
Level 1	50	–	0.74
Level 2	50	–	0.64
Level 3	50	–	0.58
Level 4	50	–	0.50
Complete	50	0.15	0.34

relevant category only. For the free search, agreement on levels 1 to 4 of the DMOZ directory is much higher, from an agreement of 0.74 on the first level, to an agreement of 0.50 on the fourth level. For the list selection, the agreement for the best matching relevant category is very similar with 0.75 at the top-level, and 0.48 at level 4.

Summarizing, from our user study we can conclude that for nearly all topics a relevant DMOZ category can be found. Categories selected in the free search are more specific than the categories from the list of suggestions. For the test persons there are no large differences between selecting categories from a list of suggestions and the free search considering speed, confidence, difficulty and personal preference. Agreement between test persons is moderate, but increases considerably when we look only at the top-level categories.

6. EXPERIMENTS

In this section we report on our experiments that exploit the topical context as retrieved from our user study.

6.1 Experimental Set-Up

To test our topical feedback approach, we use Terabyte topics 800 to 850 that have been classified by at least two test persons in our user study.

All parameters for the topical models are the same as used in the user study. However, for retrieval we do use a Porter stemmer, because our initial results indicate that stemming leads to better results. We also experimented with document length normalization, but that does not lead to any improvements. For parameter β we try values from 0 to 1 with steps of 0.1. For computational efficiency we rerank results. The run we are reranking is created by using a standard language model, with Jelinek-Mercer smoothing (λ = 0.9). We rerank the top 1,000 results.

From our user study we extract topical classifications on three levels. The deepest level topical models are the categories selected most frequently in the free search, so on any level in the directory (Free Search). The middle level consists of the categories selected most frequently from the suggested categories of levels one to four of the directory (Suggestions). We add a third classification on the top level, where one of the thirteen top level categories is picked. For the top level category we use the top category that occurs most frequently in the categories from the suggestions list (Top Level). When there is a tie between categories, we decide randomly.

Table 4: Retrieval results using topical context

Topical Context	Beta	MAP	P10
Baseline	0.0	0.2932	0.5540
Top Level	1.0	$0.0928^{\bullet}$	$0.1000^{\bullet}$
Suggestions	1.0	$0.1388^{\bullet}$	$0.2160^{\bullet}$
Free Search	1.0	$0.2179^{\circledcirc}$	$0.3640^{\circledcirc}$
Top Level	0.7	0.2937^{-}	0.5700^{-}
Suggestions	0.6	0.2984^{-}	0.5720^{-}
Free Search	0.6	$0.3238^{\bullet}$	$0.6140^{\circledcirc}$

Significance of increase or decrease over baseline according to t-test, one-tailed, at significance levels 0.05($^{\circ}$), 0.01($^{\circledcirc}$), and 0.001($^{\bullet}$).

6.2 Experimental Results

Table 4 shows the retrieval results. The baseline run does not use topical context. First, we look at how well the topical context captures the information need of the topics. As expected, when only the topical context is used ($\beta = 1.0$), results are significantly worse than the baseline. The free search categories do still perform quite reasonably, showing that the DMOZ categories can capture the information request at hand. Second, we look at combining the baseline run with topical context. In the table only the best runs are shown. Topical context using the top level categories or the suggested categories only leads to small, not significant improvements in early precision. We see that topical context on the deepest level retrieved using free search in the DMOZ directory leads to the best results with significant improvements over the baseline where no topical context is used. We show MAP and P10 over different values of β in Figure 2. The results start degrading only at a high value of β at around 0.8 or 0.9, suggesting that the topical context is quite robust.

In terms of effectiveness, there seems to be a relation with the depth in the DMOZ hierarchy. Figure 3 shows the correlation between the level of the category used as topical context, and the improvement in MAP as the result from using the free search categories as topical context. Besides the average MAP improvements per level, we added the MAP improvements per query topic.

Topical context in the form of a DMOZ category significantly improves retrieval results when the DMOZ categories are selected using free search allowing categories at any level of the directory to be selected.

7. DISCUSSION AND CONCLUSIONS

In this paper we investigated methods to get and use topical context from users where the DMOZ directory provides topic categories. We investigated two research questions, our first one being: *Can the DMOZ directory be used to effectively categorize query topics into topic categories?* We conclude that the DMOZ directory can be considered suitable to categorize query topics into cat-

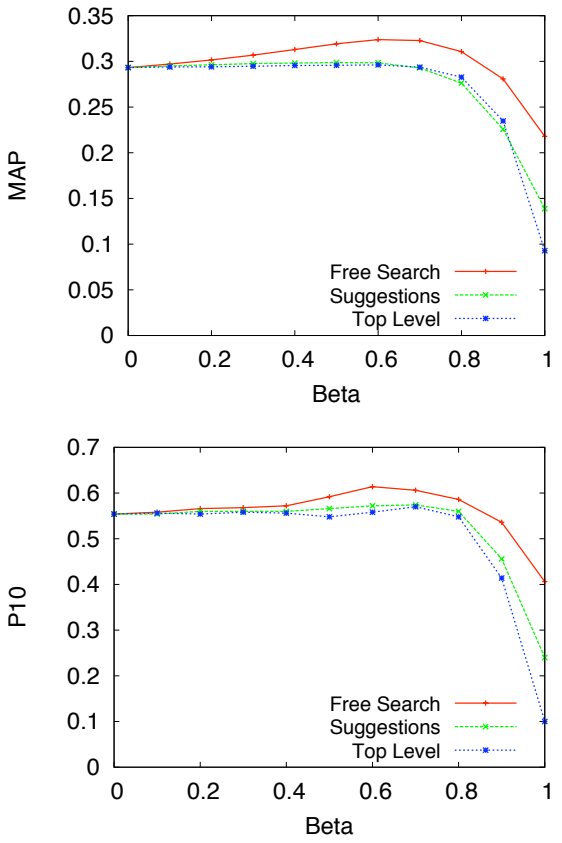


Figure 2: Topical context: MAP and P10

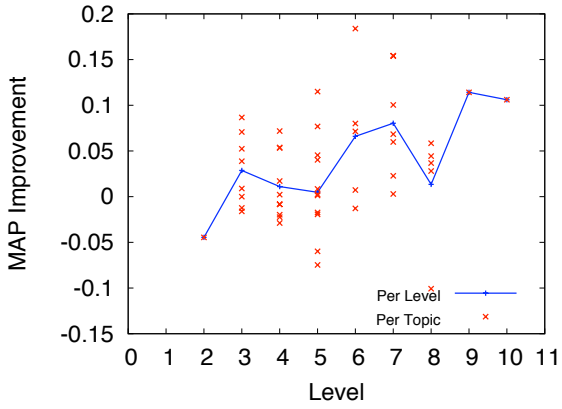


Figure 3: Correlation between level of free search topic category and MAP improvement

egories. Using either free search or the suggestions list for 98.5% of the query topics a relevant DMOZ category is found. This category can however be too broad or too specific. When test persons evaluate categories from a list of suggestions, only 19.9% of the categories is evaluated to be relevant. The relevant categories are usually too broad. For many topics, the categories till level 4 of the DMOZ category are not specific enough to categorize topics appropriately, because when we look at the categories selected by the free search, in 61% of the cases, the selected category is at level 5 or deeper.

Considering the method to use to elicit the topical context, there is no clear preference from the test persons point of view. In our set-up there is however a difference in the quality of the topic categorization. The list of suggestions only retrieves topic categories until level 4, thereby excluding a large part of the DMOZ directory. When free search is used, most often a category on level 5 is selected. Extending the automatic categorization used to produce suggestions to the fifth or a even deeper level, thus has clear potential to improve the quality of the suggestions list. Our test persons now consider evaluation of suggested categories easier, and they are also faster. It would be interesting to see if these advantages still hold when deeper level categories are also shown in the suggested categories list.

Looking at the different methods of automatic topic categorization, the title match of the query words with DMOZ category labels produces high quality suggestions, but not for many topics. Using a more lenient title match, where not all query words have to occur in the category title could provide us with more possible relevant topic categories. The categories produced by the classification of the query differ substantially from the categories produced by the classification of the top 10 documents. Differences in agreement and the coverage of query topics, are however still small. To make the list of suggestions classification of the query, the top 10 retrieved documents, and the query title match, can all three produce different useful suggestions. We do not have to choose between these methods, since users can easily review the list of suggestions and make decisions on relevance.

What is the agreement on the relevance of DMOZ categories between different test persons? Considering the test persons can choose from 590,000 categories, the lenient agreement of 0.34 for the free search is quite good. For the list based suggestions, the lenient agreement over all categories deemed relevant by any of the test persons is 0.20. A problem with the evaluation of the suggestions list is that some test persons tend to select only one or two categories, while other test persons evaluate substantially more categories as relevant, but too broad, leading to a lot of disagreement. That is, if we consider only the best matching category assigned by both judges, the lenient agreement is as high as 0.75.

Since best matching categories can be deeply nested in DMOZ, getting the initial levels of these categories right can be very important. That is, each category also represents all their ancestors' categories in the DMOZ's hierarchy. Agreement on levels 1 to 4 of the directory is much better, so at least test persons start out on the same path to a topic category. They may only in the end select different categories at different levels of granularity.

Overall, free search results in the best and most specific categories, considering agreement and coverage of the query topic. However, the categories in the suggestions list can still be improved by including more of the DMOZ hierarchy. From the test persons point of view, there is no agreement on a preference for one of the methods. So, a good option will be to use a combination of both methods so that users can decide for themselves per query how they want to select a category.

Our second research question was: *Can we use topical context to improve retrieval effectiveness?* Our experimental results show that topical context can indeed be used to improve retrieval effectiveness, but the topical categories need to be quite specific for any significant improvements. Top level categories, and the suggested categories from our list that go up to the fourth level, do not provide enough information to improve average precision. These categories could however be useful to cluster search results.

Looking at the level of the topic category in correlation to MAP improvement, we find a weak positive correlation. Deeper levels of categorization are likely to lead to better MAP improvements, but we need more data for statistical proof.

A common and effective way to improve retrieval effectiveness is to use (pseudo) relevance feedback. On this TREC data set it is found that combining topical context and pseudo relevance feedback leads to better results than applying either of them separately [10]. So while topical context alone might not outperform (pseudo) relevance feedback, their contributions to performance are complementary.

Finally, our main research question: *Can we effectively use the DMOZ directory as a source of topical context?* We can conclude that the DMOZ directory is a good option to use as a source of topic categories, since for the vast majority of query topics at least one relevant topic category is found. Two methods to elicit topical context are compared, free search on the DMOZ site to select the best category, and evaluation of a list of categories. Free search is most effective when agreement and coverage of query topics is considered. According to the test persons none of the methods is clearly better. To create the list of suggestions a combination of classification of query, top 10 retrieved documents, and a query title match can be used. Looking at retrieval effectiveness the more specific free search categories are to be preferred, since these categories are the only categories that lead to significant improvements over the baseline.

In future work we want to address the question whether automatic categorization into topic categories can also benefit retrieval. In that case no input from the user is required. So far, free search categories chosen by test persons seem to be of a better quality than suggested categories obtained by automatic categorization, but extending the automatic categorization into deeper levels of the hierarchy might lead to better results.

Acknowledgments.

We would like to thank Rongmei LI and Djoerd Hiemstra for their cooperation, and all test persons for their efforts. Rianne Kaptein was supported by the Netherlands Organization for Scientific Research (NWO, grant # 612.066.513). Jaap Kamps was supported by NWO (grants # 612.066.513, 639.072.601, and 640.-001.501).

REFERENCES

[1] L. Azzopardi, M. Girolami, and C. van Rijsbergen. Topic based language models for ad hoc information retrieval. In *IEEE International Joint Conference on Neural Networks*, pages pp. 3281–3286, Budapest, 2004.

[2] J. Bai, J.-Y. Nie, H. Bouchard, and G. Cao. Using query contexts in information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 15–22. ACM Press, New York NY, 2007.

[3] S. Buttcher, C. Clarke, and I. Soboroff. The trec 2006 terabyte track. In *The Fifteenth Text REtrieval Conference The Fifteenth Text REtrieval Conference (TREC 2006) Proceedings*, 2006.

[4] P. Chirita, W. Nejdl, R. Paiu, and C. Kohlshuetter. Using odp metadata to personalize search. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, 2005.

[5] DMOZ. The Open Directory Project, 2008. URL `http://www.dmoz.org`.

[6] T. H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference*, 2002.

[7] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, Center for Telematics and Information Technology, University of Twente, 2001.

[8] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings SIGIR 2004*, pages 178–185. ACM Press, New York NY, 2004.

[9] J. Kamps. Effective smoothing for a terabyte of text. In *The Fourteenth Text REtrieval Conference (TREC 2005)*. National Institute of Standards and Technology. NIST Special Publication, 2006.

[10] R. Kaptein, J. Kamps, R. LI, and D. Hiemstra. Experiments with positive, negative and topical relevance feedback. In *The Seventeenth Text REtrieval Conference (TREC 2008) Notebook.*, 2008.

[11] R. LI, R. Kaptein, D. Hiemstra, and J. Kamps. Exploring topic-based language models for effective web information retrieval. In *Proceedings DIR 2008*, 2008.

[12] F. Liu, C. Yu, and W. Meng. Personalized web search by mapping user queries to categories. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages pp. 558 – 565. ACM Press, New York NY, 2002.

[13] M. A. Rosso. User-based identification of web genres. *Journal of the American Society for Information Science and Technology*, 59(7): 1073–1092, 2008.

[14] J. Trajkova and S. Gauch. Improving ontology-based user profiles. In *Proceedings of RIAO 2004*, 2004.

[15] TREC. Text REtrieval Conference, 2008. `http://trec.nist.gov/`.

[16] X. Wei and W. B. Croft. Investigating retrieval performance with manually-built topic models. In *Proceedings of RIAO 2007 - 8th Conference - Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, 2007.

[17] Wikipedia. The Free Encyclopedia, 2008. URL: `http://www.wikipedia.org`.

[18] Yahoo! Directory, 2008. URL `http://search.yahoo.com/dir`.

[19] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 49–56. ACM Press, 2001.

Business Specific Online Information Extraction from German Websites

Yeong Su Lee
CIS, University of Munich
Oettingenstr. 67
D-80538 Munich, Germany
yeong@cis.uni-muenchen.de

Michaela Geierhos
CIS, University of Munich
Oettingenstr. 67
D-80538 Munich, Germany
micha@cis.uni-muenchen.de

ABSTRACT

This paper presents a system that uses the domain name of a German business website to locate its information pages (e.g. company profile, contact page, imprint) and then identifies business specific information. We therefore concentrate on the extraction of characteristic vocabulary like company names, addresses, contact details, CEOs, etc. Above all, we interpret the HTML structure of documents and analyze some contextual facts to transform the unstructured web pages into structured forms. Our approach is quite robust in variability of the DOM, upgradeable and keeps data up-to-date. The evaluation experiments show high efficiency of information access to the generated data. Hence, the developed technique is adaptive to non-German websites with slight language-specific modifications, and experimental results on real-life websites confirm the feasibility of the approach.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process; I.2.7 [**Natural Language Processing**]: Text analysis

General Terms

company search, information extraction, sublanguage

1. INTRODUCTION

With the expansion of the Web, the demand for targeted information extraction is continuously growing. There are many services on the Web providing industry sector information or performing job search tasks. For these purposes, the data used must be first manually collected and therefore features several sources of error, e.g. spelling mistakes, incomplete database entries, etc. Moreover, this process is extremely time-consuming and updating the data then requires a rollback of the full process. Automating these tasks will help to extract the business specific information quickly and maintain the data up-to-date.

The standard approach of business-related information retrieval disregards the relationship between the domain name and organization-specific content of a website, but concentrates on the structural aspect of company information [2]. Only a few studies restrict the information extraction task to certain domain names [8, 9, 14]. They extract company profiles by limiting their research on locating products and other features while analyzing the format of HTML tables for structured data and trying to find the phrase patterns for unstructured texts [8]. Others examine the presentation ontology for extracting organization-specific data such as contact details and product information concentrating on the differences in the presentation manner of formatted company profiles versus plain text profiles [9]. But company information extraction can also be extended to different resources and incorporates meta tags as well as plain texts and structured data [14].

As the Web keeps evolving, of course, every new website will uncover new ways that people encode the information. That way, other scientists concentrate on linguistic analysis of web pages and disregard the main characteristic advantage of the HTML structure. They investigate, for example, information extraction techniques for company details and job offers on the Web. These methods consider the relevance of the domain name, but only exploit the local characteristics of the text [1]. They therefore process in two steps: first HTML stripping and then applying local grammars [5] (recursive transition networks) on plain texts to transform unstructured web pages into structured forms. Manually encoding morphosyntactic rules for extracting the information seems doomed to be a never-ending process, but evaluation experiments show high values of precision and recall.

Our starting point of a solution is the structured nature of data. In contrast to a general search scenario, company search can be seen as a slot-filling process. The indexing task is then to detect attribute-value pairs in the HTML documents and make them accessible. At this point, we are interested in the extraction of all organization-specific data being related to the web site's domain name (secondary level domain). Obligatory elements, such as the company name combined with a highly restrictive domain vocabulary, make it possible to discover the logic of an information page that can then be integrated into a relational structure. As our studies during this research were limited to the German Web, the investigated language was German.

The paper is structured as follows. In the next section we introduce the concepts and terms used in the paper. Section 3 presents an overview of the system architecture. In Section 4 the analysis of the information page is further detailed and Section 5 evaluates the performance of the system and shows promising results of precision (99.1%) and recall (91.3%). The conclusion comments on practical implications of the given approach and the directions of future work.

2. DEFINITION OF TERMS

Terms that are used throughout this paper in various contexts and that have a particular usage have to be clearly defined.

2.1 Business specific information

Business specific IE differs from the record extraction or entity recognition because the information must be examined with respect to the domain name and estimated how valuable it may be.

Definition 1 (Business specific information)
Business specific information contains the relational facts concerning the domain name.

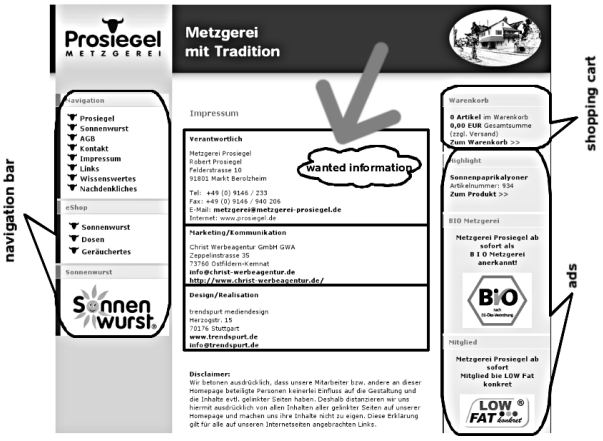


Figure 1: Example of an information page

In order to illustrate what kind of information is relevant according to the domain name, one information page is shown in Figure 1. The left section contains the navigation bar, the right one a shopping cart and advertisements, and the center is divided into three information records: The first contains the domain relevant information we are interested in. The second also appears somehow relevant but is about specialized marketing and the third names the web designer.

2.2 Minimal data region

A group of data records that contains descriptions of a set of similar objects are typically presented in a particular region of a page (...) Such a region is called a ***data region****.* [10]

We can identify the region of the information bit with keywords or phrases heading the respective record. In our example (cf. Figure 1), the heading keyword for the relevant information is *"Verantwortlich" (responsible)*, for the marketing information it is *"Marketing/Kommunikation" (marketing/communication)*, and for the web designer record it is *"Design/Realisation" (design/ realization)*.

But we have to limit the data record containing information somehow focused on the domain name. In contrast to other approaches [10] we are not interested in locating data records of maximum length, we want to determine the "minimal data region" for an information bit (cf. Section 4).

Definition 2 (Minimal data region)
A minimal data region with respect to the business specific information is the smallest HTML tag region where most of the wanted information bits are located.

2.3 Sublanguages on the Web

Definition 3 (Web sublanguage)
Sublanguages are specialized language subsets, which are distinguished by the special vocabulary and grammar from the general language [6, 7]. With respect to the Web, a sublanguage is characterized by a certain number of phrases or a grammar and special vocabulary [4], e.g. "Impressum" (imprint).

Web sublanguages occur on the home page of a website as well as on its information page. Regarding the home page we analyze the anchor texts that lead to the information page (cf. Figure 2). But the variety of organization-specific standard phrases (*frozen expressions*) that frequently emerge on information pages are clustered into *attribute classes* during the training step of our system. For instance, the class *"Provider"* contains about 140 specialized words and phrases (*attributes*), e.g. *"Anbieter i.S.d. TDG/MDStV" (Provider in terms of TDG/MDStV)* (cf. Table 1).

Attribute Class	Quantity	Vocabulary
company name	99	*Anbieter, Firmenbezeichnung*
phone no.	25	*Fon, Tel, Tel + Fax*
fax no.	7	*Fax, Faxnummer, Telefax*
mobile no.	13	*mob, mobil, unterwegs*
email	16	*Mail, E-Mail, m@il*
CEO	23	*CEO, Geschäftsführer*
business owner	16	*Inh, Inhaber, owner*
contact person	10	*Ansprechpartner, Kontaktperson*
chairman	23	*chairman, Leiter, Vorsitzender*
management board	4	*Vorstand, Geschäftsführender Vorstand*
VAT ID	97	*UID, UST-ID-NR, Umsatzsteueridentnr.*
tax no.	25	*St. Nr., Steuernr, Umsatzsteuer Nr.*
register no.	22	*Handelsnr., Registernummer*
local court	28	*AG, Amtsgericht*
tax office	4	*FA, Finanzamt*

Table 1: Overview of attribute classes pertinent to business websites

2.4 Business specific information extraction

Definition 4 (Business specific IE)
Business specific information extraction is concerned with the automatic extraction of the relation between a domain name and an information set consisting of attribute-value pairs.

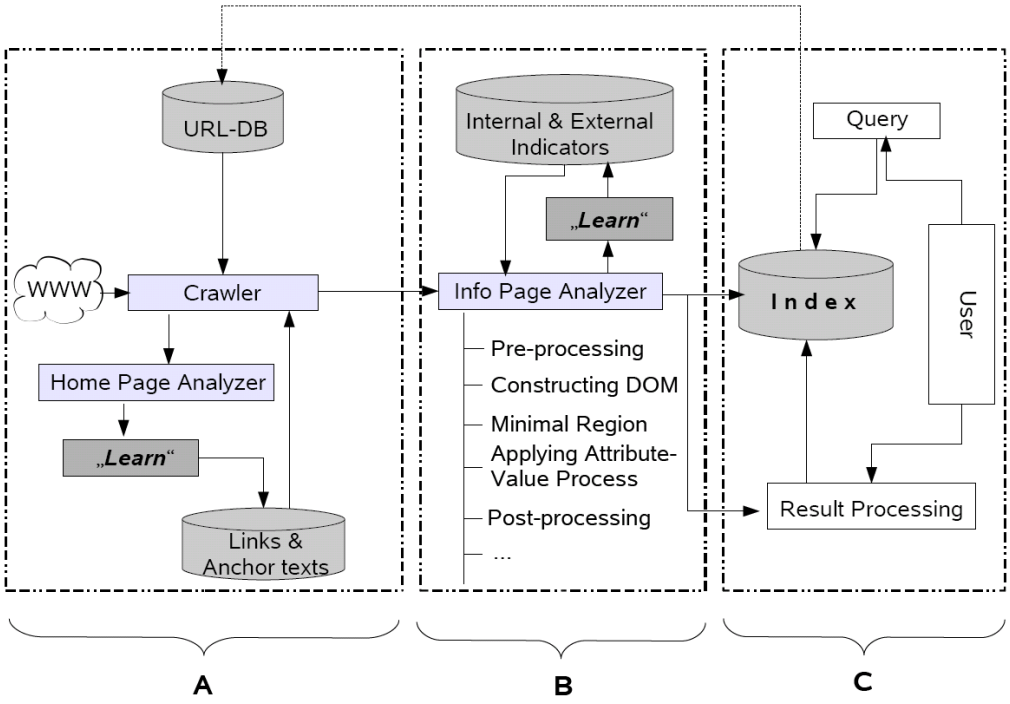


Figure 2: Overview of the system architecture of *ACIET*

3. SYSTEM ARCHITECTURE

Figure 2 shows the elements of our system[1] to extract business specific data from information pages of German websites. This process expects as input a set of URLs preclassified as business websites.

The architecture is based on two interactive modules to establish a relational database storing company information and providing a query module:

- **A** Localization of information pages on the Web
- **B** Document analysis and information extraction
- **C** Query processing

Our system `ACIET` (**A**utomatic **C**ompany **I**nformation **E**xtraction **T**ool) automates the extraction process of organization-specific information on the Web and works therefore in two steps:

In the first stage (**A**), a focused crawler is fed with URLs stored in a database and fetches the demanded websites. This step is performed by the "home page analyzer" [2]. Our system will follow the anchor tags leading to the information page and retrieve the document.

[1]For research and test purposes the prototype of our system is available at `http://www.cis.uni-muenchen.de/~yeong/ADDR_Finder/addr_finder_de_v12.html`.

[2]For classification purposes, it can also extract the structural and textual features of a website by category. But at present we are only focused on the extraction process and suppose that our crawler input exclusively consists of business websites.

During the second stage (**B**), the information page is sent to a module called "info analyzer" to study the HTML content and extract the searched information bits. It thereby exploits the internal structure of named entities and uses sublanguage-specific contexts – attribute classes (cf. Section 2.3) to identify the attribute-value pairs. In difference to other systems the form filling process is fully automatized. From a document recognized as an information page by the system (part **A**) we extract all business specific information to fill a form that is presented in Table 2.

Example of a company info form	
company name	Metzgerei Prosiegel
street	Felderstraße 10
zip code	91801
city	Markt Berolzheim
phone no.	(09146) 233
fax no.	(09146) 940206
email	metzgerei@metzgerei-prosiegel.de

Table 2: Business specific information of Fig. 1

For the transformation of the initial HTML-document into the form schema we need different operations shown in Figure 2 (part **B**).

An interaction by the user is provided in part **C** (cf. Figure 2). There, the user can query the database and supervise which information bit extracted by `ACIET` will be added to the index.

4. INFORMATION PAGE ANALYZER

Given an information page, the preprocessing starts with analyzing the frame structure and existing javascript. Before creating an expressive DOM structure [10, 11], the HTML file has to be validated and if necessary corrected. This step is done by the open source unix tool `tidy`[3]. Now our system is able to locate the minimal data region (for more details see Section 4.1) surrounded by certain HTML tags containing the information record searched for. During a depth-first traversal of the DOM tree, the wanted subtree can be isolated according to the headings of the data record, e.g. *"Herausgeber" (publisher)*, *"Betreiber" (operator)* or *"Anbieter" (provider)*. Since we disregard domain name irrelevant information, we will work further on with a pruned DOM tree. After identifying the minimal data region, all information bits relevant to the domain name are extracted by the attribute-value process (for more details see Section 4.2) with respect to external contexts and internal features. Our system considers about 20 attribute classes and searches their values on the information page of business websites [17]: *company name, address, phone and fax number, e-mail, CEO, management board, domain owner, contact person, register court, financial office, register number, value added tax number (VAT ID)*, etc.

4.1 Detecting the minimal data region

As already shown in Figure 1, an imprint page contains lots of noisy and irrelevant data. In order to determine the minimal data region, we pursue three strategies:

1. Depth-first traversal of the DOM tree to locate the data region of the information bit searched for.

2. Isolation of subtrees containing information bits according to specified headings and pruning of the DOM tree by deleting domain name irrelevant data.[4]

3. Detecting the minimal data region with respect to predefined attribute classes (*"phone number"*, *"fax number"* and *"VAT ID"*).

This method works perfectly (see precision and recall in Table 4) and efficiently due to the minimal text length of the data region. That way, ambiguities arising by reason of multiple contexts are eliminated before they emerge.

4.2 Attribute-value process

Detecting the minimal data region limits the search areas in the DOM tree, but does not resolve any ambiguities. If we use, for example, a pattern-based approach to determine a phone number, the same regular expression can also match a fax number. Now we have to assign the correct values to the attributes according to close-by HTML content information provided by the DOM tree.

[3] `http://tidy.sourceforge.net`

[4] We are now able to delete all subtrees captioned by any negative heading (e.g. *"Design" (design), "Realisierung" (realization), "Umsetzung" (implementation), "Web-Hosting" (web hosting)*) from the document object model. That way, this pruning step isolates the business specific subtrees and even eliminates "negative-headed" regions of the tree nested in subtrees preceded by positive titles.

The recognition of person names causes similar problems: Searching for names on the DOM tree facilitates their localization because these strings are delimited by the HTML tags surrounding the entry. The internal structure of the person name will be characterized by a rule-based method, e.g. a non-left-recursive definite clause grammar. But to discover the person's role, we have to rely on the fact that names occur close to context words hinting on the corresponding attribute classes.

That way, the named entity recognition can profit by the HTML structure which refines the search space. To distinguish the person's function, the "value" (person name) has to be extracted together with its "attribute" (attribute-value pair). All known attributes were collected during the training stage of our system and compiled into a trie. Moreover, unknown context words can also be correctly attributed by approximate matching with `agrep`[5] [15].

The most remarkable advantage of the attribute-value process is the fact that for the named entity recognition, no large lexicon is required. Thus, the identification of person names is much faster than by a lexicon-based approach. How external and internal indicators work together to guarantee such a success will be discussed in the next section.

4.2.1 Internal and external indicators for NER

Internal evidence *is derived from within the sequence of words that comprise the name. (...) By contrast,* ***external evidence*** *is the classificatory criteria provided by the context in which a name appears.* [12].

Mikheev et al. (1999) [13] observed the importance of internal and external evidences for the named entity recognition (NER) at the MUC-7 conference. They experimented with several lexicon sizes and discovered that a large comprehensive lexicon cannot improve considerably the precision or recall of a NER system.

Hence, we also pursue this strategy and compile the internal and external indicators into the corresponding attribute classes. Some examples for external indicators obtained during the training phase are shown in Table 1. Moreover, the list of indicators is open-ended and managed within different files – a sublist per attribute class.

There are two different types of internal indicators: vocabulary lists and regular expressions for digits like phone or fax number. With regard to company name recognition, we can benefit, for example, from 35 legal forms, 130 business types, 400 job titles, and some typical affixes of company names.

4.2.2 Creating an expressive DOM structure

Since the DOM tree does not reflect the fundamental characteristics of all HTML tags, we will cluster the HTML tags by their formatting function.

We therefore divide the HTML tags in six groups: *character, heading, block, list, table, and image elements.*

[5] cf. `http://www.tgries.de/agrep`

It is quite obvious that some tags within other tag regions might loose the differentiating property. That way, this deletion of HTML tags helps us to interpret the role of an HTML element within the whole DOM tree and to ignore pointless misplaced elements.

4.2.3 Recognition of attribute-value pairs in tables

About 70% of the information pages used during the training period encode business specific data in HTML tables. Since those tables totally differ in structure [3], their recognition will cause some problems if we always pursue the strategy to extract the value in the right context of the attribute.

During the attribute-value process, we don't really have to recognize the table type (cf. Figure 3). Instead, we apply the attribute-value process directly to the table cells.

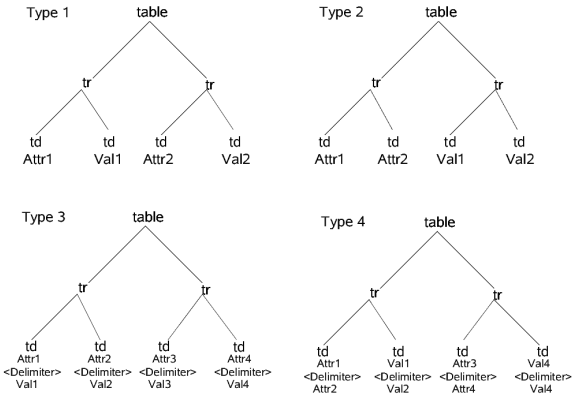


Figure 3: Different types of HTML tables containing attribute-value pairs

The extraction of attribute-value pairs in tables of type 1 and 3 seems trivial. If an instance for one of our predefined attribute classes is found, according to type 1, the cell in the next column will be scanned for the corresponding value of the attribute. For type 3, given an attribute separated by at least one delimiter the search for the value can be performed on a single column because both – attribute and value – are located together in the same cell.

However, we have to face a minor difficulty for type 2 and 4. The structure of type 2 shows that attributes and values are separated by the <tr>-tag and span over two lines. Therefore, the search algorithm has to be adapted to the new situation: After locating the first attribute, the cell in the next column is tested for values or further attributes. This recursive step will be repeated until the corresponding value is identified.

Type 4 is very complex in comparison to the other table structures. Since each cell contains several pieces of information separated by at least one delimiter, we will manage the data by a two-dimensional array. The algorithm therefore implemented is shown in Figure 4. One problem occurring quite often is that close-by cells do not contain the same number of delimiters. Thus, a complete scan of the cell divided by the delimiters is necessary and this step has to be repeated until the correct value can be assigned to the corresponding attribute.

Pseudo-algorithm of the attribute-value process for table type 4

```
1. tds = Find tds; // Array of columns
2. anz_tds = Number of tds; // Number of columns
3. td_delimiters = break each column by <Delimiter>; // Array of Delimiters of columns
4. for ( i = 0; i <anz_tds - 1; i++ ) // for each column except the last
5.     anz_delimiters_td = Number of Delimiters of td_delimiters[i];
                                        // Number of Delimiters of the concerned column
6.     for ( j = 0; j < anz_delimiters_td; j++ ) // For each text field
7.         next unless length td_delimiters[i][j]; // jump, if empty ist
8.         td_text = td_delimiters[i][j];
9.         for each Attribute_class of already classsified attribute classes
10.            if td_text is Element of concerned Attribute_class
11.               anz_delimiters_next_td = Number of Delimiters of td_delimiters[i+1];
                                        // Number of Delimiters of  next column
12.               for ( k = 0; k <anz_delimiters_next_td; k++ ) // for the next column
13.                  next unless length td_delimiters[i+1][j];
14.                  if td_delimiters[i+1][k] is Wert_text of concerned Attribut class
15.                     Extract (td_text,  td_delimiters[i+1][k]);
16.                     Delete (td_text,  td_delimiters[i+1][k]);
17.                     Break for inner for-loop
18.               endif
19.            endfor
20.        endif
21.    endfor
22. endfor
23. endfor
```

Figure 4: Pseudo-algorithm to identify the attribute-value pairs in table type 4

4.2.4 Other structures

Subtrees of the DOM other than HTML tables are also traversed by the attribute-value-process. After locating an attribute, the corresponding value has to be searched within the next HTML tag region or within the string containing an instance of the attribute class and at least one delimiter. Our system will limit the search area in the DOM tree by a pair of attributes and then go through the HTML content elements separated by tags or delimiters string by string.

Moreover, the contextual information can also be used to extract company names from the HTML document. There is often some legal notification on the information page hinting on the domain operator, e.g.

- *Publisher of this website is the*
- *Service provider of these pages is the*
- *This is the joint internet appearence of the company*
- *assumes no liability*
- *can not guarantee for the completeness*
- *accepts no responsibility for the correctness and completeness*

4.3 Postprocessing

All extracted information bits have to be normalized afterwards to guarantee the data consistency. The normalization process affects the following attribute classes:

- company name, legal form, register number
- address: street, zip code, city
- contact: phone and fax number, email
- person name
- legal notification: tax number and VAT ID

The legal form within a recognized German company name usually indicates the register department. Some legal forms like *"GbR, KG"* are registerd in the department **"A"**, while others like *"GmbH, AG"* in the department **"B"**. But the department is not always given correctly. Hence, the coherence between recognized legal form and register department must be checked in order to assign the right department to the register number.

The postprocessed data is organized in lexica for the zip code, city and area code.

In Table 2 we already showed an example of an automatically created company information form of the business website `www.prosiegel.de`. The values filled in these slots are normalized according to the above mentioned techniques. After locating the attribue-value pairs, the values are tagged by the corresponding attribute classes on the web page (cf. Figure 5).

Impressum

Verantwortlich [FN]Metzgerei Prosiegel[/FN]
Robert Prosiegel
[STR]Felderstrasse 10 [/STR]
[PLZ]91801[/PLZ]
[ORT]Markt Berolzheim [/ORT]
Tel[TEL]: + 49 (0) 9146 / 233[/TEL]
Fax[FAX]: + 49 (0) 9146 / 940 206[/FAX]
E-Mail: [EMAIL]metzgerei@metzgerei-prosiegel.de[/EMAIL]
Internet: www.prosiegel.de

Figure 5: Annotated company information of Fig. 1

Since information pages are set up from humans for humans, some spelling mistakes can also occur there and have to be corrected, e.g. *"Felderstrasse"* must be *"Felderstraße"*.

In order to get a uniform phone number, we have to delete all non-digits and the country code, match the longest area code provided by the lexicon and separate the number into the area code and direct outward dialing sequence. So the phone number mentioned in Figure 5 will be transformed to *"(09146) 233"* and the fax number to *"(09146) 940206"* (cf. Table 3).

street	Felderstrasse 10	Felderstraße 10
phone no.	+49 (0) 9146 / 233	(09146) 233
fax no.	+49 (0) 9146 / 940 206	(09146) 940206

Table 3: Normalized attribute values of Fig. 5

Person names often appear as uncapitalized sequences. In this case the uniform format can be reconsructed by the postprocessing.

With respect to the tax number and VAT ID, the postprocessing is indispensable. Not always information is given according to the standard scheme of the tax number and VAT ID. Given an external indicator *(attribute)* hinting on a VAT ID, our system will expect this number *(value)* to be a VAT ID. But instead of a VAT ID, for example, the tax number follows: *"Umsatzsteuer-Identifikationsnummer gemäß 27a Umsatzsteuergesetz: DE 053-116-00763"*. The postprocessing step now allows our system to adjust its assumption: The given code *DE 053-116-00763* is not conform to a standardized VAT ID. So we replace the hyphen (-) by a slash (/) and get the valid scheme of a German tax number. During the evaluation scenario, our system correctly identified the tax number in 13 cases, although the local context refers to a VAT ID.

5. EXPERIMENTAL EVALUATION

To evaluate the quality of our system with regard to the recognition of information bits indicating business specific data, we designed a small, manually verified test corpus composed of approximately 150 SLDs (websites).

5.1 Test-data design

For creating this test base, our system[6] was fed with 924 SLDs picked up randomly by the focused crawler. Among these, 478 SLDs were determined to be appropriate candidates for company websites.[7] The evaluation process was then limited to every third SLD of the candidate set and these 159 SLDs were checked afterwards by visiting the sites with a web browser. As there existed several copies of some SLDs and others were no longer available on the Web, only 150 SLDs remained for test purposes.

5.2 Evaluation results

Table 4 shows promising results of precision (99.1 % on average) and recall (91.3 % on average) considering the recognition of entities typically found in information pages of business websites. The experimental evaluation presented in this paper is limited to 16 information bits not counting those that have less than 10 instances on the test data.

5.3 Discussion

Needless to say, the evaluation results displayed in Table 4 show more lack of recall than precision. However, we want to discuss the reasons of it.

5.3.1 Lack of precision

Only three of totally 16 information bits vary in precision:

Company Name. Due to the fact that no headings are given, the system will choose the first company name candidate. This decision is based on the higher probability of company names appearing before web design or host details. But we have to admit that in some cases this kind of heuristics does not work and drops the precision to 96.3%.[8]

[6]For research and test purposes the prototype of our system is available at `http://www.cis.uni-muenchen.de/~yeong/ADDR_Finder/addr_finder_de_v12.html`.

[7]This step was performed by an external tool – a classificator for business websites not described here.

[8]For the URL `http://www.bergener-rathaus-reisebuero.de/shared/impressum.html`, for example, the company extracted from the information page is *"2000 RT-Reisen GmbH"*, but it should actually be *"Reisebüro am Bergener Rathaus"*.

Extracted Type of Information	Total	Extracted	Correct	Precision	Recall
company name	150	134	129	96.3%	86.0%
street	150	149	147	98.6%	98.0%
zip code	150	150	150	100%	100%
city	150	150	150	100%	100%
phone no.	137	135	134	99.2%	97.8%
fax no.	125	124	124	100%	99.2%
mobile no.	13	13	13	100%	100%
email	126	124	124	100%	98.4%
VAT ID	73	72	72	100%	98.6%
tax no.	25	22	22	100%	88.0%
CEO	39	28	28	100%	71.7%
business owner	24	21	21	100%	87.5%
responsible person	33	24	24	100%	72.7%
authorized person	12	11	11	100%	91.6%
local court	44	38	38	100%	86.3%
register no.	45	38	38	100%	84.4%
On average				99.1%	91.3%

Table 4: Evaluation results gained on the test SLDs

Street. Our grammar-based approach expects certain suffixes to recognize street names, e.g. *"-straße" (street), "-gasse" (lane), "-weg" (road),* etc. Without such an indicator a street name will not be identified. Only streets ending on these special suffixes are extracted, no matter where they are located on the information page. But it happens to be the false name if more than one street name is given and the right one does not have such a suffix. This lowers the precision to 98.6%.[9]

Phone No. After locating a phone number, it is normalized by the system to a consistent format. A number like *02851/8000+6200*[10] is then transformed to *(02851) 80006200*. But the deletion of "+" is not correct. That way, the plus expresses an alternate phone number – a kind of ellipsis – which will not be resolved and two numbers are merged to one single number. This error appeared only once, so that the precision is not strongly influenced (99.2%).

5.3.2 Lack of recall

13 of totally 16 information bits vary in recall, but only two go below the 80%-boundary. The reasons for their incomplete or none-recognition are due to

- flash animations, javascript and images protecting the piece of information searched for.
- missing external indicators on information pages, e.g. *Tel., Fax, E-Mail*
- missing syntactic rules that describe the internal structures of streets, etc.
- textual representations of phone numbers, e.g. *0700 TEATRON*
- informal specification of tax numbers, register numbers, etc.

These types of errors cause some malfunction in the system. Thus, we go into detail for those informations bits with recall values between 70% and 90%:

Company Name. The recognition of company names failed at 26 company names (a recall of 86.0%). On the one hand, this malfunction is caused by flash animations or images[11] hiding the piece of information searched for. On the other hand, some SLDs lead to websites encrypting the information presented there. Going to the start page of such sites, an intro page in the form of a full-screen image[12] waits for a reaction of the user. After clicking on a button with a pointless description, the user gets the chance to reach the navigation page. This kind of "scavenger hunt" makes it impossible to find the company name. Moreover, missing internal and external indicators prevent the correct identification of company names on websites.

Tax No./Register No. Both of them are standardized numbers. Although their syntactic structure is mandatory, these numbers could be written in slightly different forms. Sometimes the license plate code, e.g. *HH*, is used as prefix of the register number: *"Handelsregisternr.: HH 100042 Hamburg"* In place of *HH* our system expects the abbreviations *HRA* for partnership, self-employed and small business or *HRB* for corporation. However, the external indicator *"Handelsregisternr."* hints on a valid register number appearing afterwards, the left context can mislead our system and prevent the recognition of an informal specified register number (88.0%) as well as it disregards this kind of variation for the tax numbers (84.4%).

CEO/Owner/Responsible Person. As shown in Table 4 the recall for the identification of person names is lower than for the recognition of other information bits. Due to missing contexts hinting on person names and very strict regular expressions describing their internal structure, the use of additional information within the names precludes their complete localization. Some infixes like *"Architekt"* or *"Biol."* specifying the profession of a person have not been considered in this syntactic position yet. These words or abbreviations are usually situated between the academic degree and the person name:

Dipl.-Ing. ***Architekt*** *Christian Stanitzeck*
*Dipl.-**Biol.** Elek Szabo*

At present a different order of academic degree and job descriptor is matched by the grammar, e.g. *"Architekt Dipl.-Ing. Christian Stanitzeck"*. In a revised version of our grammar for person names, these features will be considered. But for the moment, this lack of accuracy reduces the recall for the automatic recognition of CEOs to 71.7%. Assuming that website owners are less frequently named than CEOs or responsible persons (cf. Table 4), the partially identification of the corresponding names behaves very similar to the owners (87.5%) and responsible persons (72.7%).

[9]For the URL `http://www.gestuet-schlossberg.de/deutsch/impressum.php` our system located the street name *"Ridlerstraße 31 B"*, but it should actually be *"Zachow 5"* which is not matched by the grammar.

[10]The phone number is taken from `http://www.pieper-landtechnik.de/seiten/impressum.html`.

[11]e.g. `http://www.hardmedia.de/`

[12]e.g. `http://www.koerperkult.de`

Local Court. This piece of information often appears in conjunction with the register number. In that case, there are no external indicators telling us that the name of a city stands for the local district court. Within an informal specified register number, e.g. *"HH 100042 Hamburg"*, this kind of metonymy will not be discovered and therefore lowers the recall rate (84.4%).

6. CONCLUSION

We presented an integrated platform to enable business specific information extraction on the Web. Though we also gave an overview on the localization of information pages on the Web, the main focus in this paper lies on document analysis and business specific information extraction. The core technique to automatically extract structured information is the attribute-value process and use of internal and external indicators hinting on the demanded information. The evaluation on the test SLDs shows excellent results for the proposed approach.

Though the linguistic descriptors and the examples of business information pages refer to the German Web, the methods are generalizable for other languages easily applicable to other countries' websites. The system expects the national specific variation of the information format and corresponding internal and external indicators. The integrated file management system can facilitate the maintenance of these indicators.

Even though every new website will uncover new ways that people encode the information, the success of our extraction method will not be affected by changing HTML structures. Tests showed that variations in web content and DOM tree do not influence the attribute-value process. Since our system relies on linguistic resources (e.g. specialized vocabulary), exhaustive studies of context information and a weighted, local interpretation of the HML tags, we can present a quite robust application.

Moreover, our system `ACIET` can be extended to integrate further text analysis tools which extract, for example, the activities of companies or their production processes.

7. REFERENCES

[1] S. Bsiri, M. Geierhos, and C. Ringlstetter. Structuring job search via local grammars. *Advances in Natural Language Processing and Applications. Research in Computing Science (RCS)*, 33:201–212, 2008.

[2] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan. A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428, 2006.

[3] G. N. David W. Embley, Daniel Lopresti. Notes on contemporary table recognition. In *Proceedings. Document Analysis Systems VII, 7th International Workshop, DAS 2006*, volume 3872, pages 164–175. Springer, Berlin, 2006.

[4] R. Grishman. Adaptive information extraction and sublanguage analysis. In *Proceedings of Workshop on Adaptive Text Extraction and Mining at Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, USA, 2001.

[5] M. Gross. The Construction of Local Grammars. In E. Roche and Y. Schabès, editors, *Finite-State Language Processing*, pages 329–354. Language, Speech, and Communication, Cambridge, Mass.: MIT Press, 1997.

[6] Z. S. Harris. Mathematical Structures of Language. *Interscience Tracts in Pure and Applied Mathematics*, 21:152–156, 1968.

[7] Z. S. Harris. Language and Information. *Bampton Lectures in America*, 28:33–56, 1988.

[8] S. Krötzsch and D. Rösner. Ontology based extraction of company profiles. In *Proceedings of the 2nd International Workshop on Databases, Documents, and Information Fusion*, Karlsruhe, Germany, July 2002.

[9] M. Labský and V. Svátek. On the design and exploitation of presentation ontologies for information extraction. In *ESWC'06 Workhshop on Mastering the Gap: From Information Extraction to Semantic Representation*, Budva, Montenegro, June 2006.

[10] B. Liu, R. Grossman, and Y. Zhai. Mining data records in web pages. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 601–606, Washington, D.C., USA, 2003.

[11] W. Liu, X. Meng, and W. Meng. Vision-based web data records extraction. In *Ninth International Workshop on the Web and Databases (WebDB 2006)*, pages 20–25, Chicago, USA, June 2006.

[12] D. McDonald. Internal and external evidence in the identification and semantic categorization of proper names. In B. Boguraev and J. Pustejovsky, editors, *Corpus processing for lexical acquisition*, pages 21–39. MIT Press, Cambridge, MA, USA, 1996.

[13] A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. In *In Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, 1999.

[14] V. Svátek, P. Berka, M. Kavalec, J. Kosek, and V. Vavra. Discovering company descriptions on the web by multiway analysis. In *New Trends in Intelligent Information Processing and Web Mining (IIPWM'03)*, Zakopane, Poland, 2003. Springer-Verlag, Advances in Soft Computing series.

[15] S. Wu and U. Manber. Agrep – a fast approximate pattern-matching tool. In *Proceedings USENIX Winter 1992 Technical Conference*, pages 153–162, San Francisco, CA, USA, 1992.

[16] M. Yoshida, K. Torisawa, and J. Tsujii. Extracting attributes and their values from web pages. In A. Antonacopoulos and J. Hu, editors, *Web Document Analysis: Challenges and Oppertunities*, pages 179–200. World Scientific, London, 2003.

[17] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. Simultaneous record detection and attribute labeling in web data extraction. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 494–503, Philadelphia, PA, USA, 2006.

Using Language Modeling for Spam Detection in Social Reference Manager Websites

Toine Bogers
ILK / Tilburg centre for Creative Computing
Tilburg University
P.O. Box 90153, 5000 LE
Tilburg, The Netherlands
A.M.Bogers@uvt.nl

Antal van den Bosch
ILK / Tilburg centre for Creative Computing
Tilburg University
P.O. Box 90153, 5000 LE
Tilburg, The Netherlands
Antal.vdnBosch@uvt.nl

ABSTRACT

We present an adversarial information retrieval approach to the automatic detection of spam content in social bookmarking websites. Our approach is based on the intuitive notion that similar users and posts use similar language. We detect malicious users on the basis of a similarity function that adopts language modeling at two different levels of granularity: at the level of individual posts, and at an aggregated user level, where all posts of one user are merged into a single profile. We evaluate our approach on two spam-annotated data sets representing snapshots of the social bookmarking websites CiteULike and BibSonomy. We find that our approach achieves promising results across data sets, with AUC scores ranging from 0.92 to 0.96.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.4 [**Information Systems Applications**]: H.4.2 Types of Systems; H.4.m Miscellaneous

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Adversarial IR, spam detection, social bookmarking, language modeling

1. INTRODUCTION

The term 'spam' was originally used to refer to the abuse of electronic messaging systems that started in the mid-1990s on Usenet newsgroups, and quickly crossed over to e-mail messaging. According to conservative estimates, in the first half of 2007 around 85% of all e-mail sent in the world was spam[1]. The notion of spam is subjective by nature, but we define it here as content designed to mislead, and that the legitimate users of a system, site, or service therefore do not wish to receive. Motivation for spamming can range from advertising and self-promotion to disruption and disparagement of competitors [5]. Spamming is economically viable because the barrier for entry into systems open to user-generated content is generally low, and because it requires virtually no operating costs beyond the management of the automatic spamming software. In addition, it is often difficult to hold spammers accountable for their behavior.

Any system that relies on user-generated content is vulnerable to spam in one form or another. Search engines, for instance, suffer increasingly from so-called *spamdexing* attempts with content especially created to trick search engines into giving certain pages a higher ranking than they deserve [4]. Spam comments are also becoming an increasing problem for websites that allow users to post reactions to content, such as blogs and video and photo sharing websites [12]. Finally, Wikpedia—another system focused around user-generated content—has also seen an increase in research interest in automatic approaches to spam detection [15].

By analogy, the relatively recent phenomenon of social websites and social bookmarking services has become an increasingly popular part of the Web, but their focus on user-generated content also makes them vulnerable to spam, threatening their openness, interactivity, and usefulness [5]. In this paper, we focus on how we can detect spam in social bookmarking systems. Our approach to spam detection is based on the intuitive notion that spam users are likely to use different language than 'legitimate' users when posting content to a social bookmarking system. We detect new spam users in the system by first ranking all the known users in the system by the KL-divergence of the language models of their posts—separately per post as well as merged into user profiles—and the language model of the new user or post. We then look at the spam labels assigned to the most similar users in the system to predict a spam label for the new user. We test our approach on two spam-annotated data sets, based on BibSonomy[2] and CiteULike[3], two so-called *social reference managers* that allow users to store and manage their reference list of scientific articles online.

The paper is structured as follows. We start by reviewing the related work in the next section, followed by a description of the task and the data sets, our pre-processing steps, and our evaluation setup in Section 3. In Section 4 we describe our spam detection approach; in Section 5 we report on our results. We conclude our paper by discussing our findings in Section 6 and listing possible future work in Section 7.

[1] http://www.maawg.org/about/MAAWG20072Q_Metrics_Report.pdf

[2] http://www.bibsonomy.org

[3] http://www.citeulike.org

DIR 2009 February 2-3, 2009, Enschede, the Netherlands.

2. RELATED WORK

The issue of spam in social bookmarking services has received relatively little attention so far. Heymann et al. (2007) were the first to examine the relationship between spam and social bookmarking in detail [5], classifying the anti-spam strategies commonly adopted in practice into three different categories: *prevention*, *demotion*, and *detection*. *Prevention-based* approaches are aimed at making it difficult to contribute spam content to the social bookmarking system by restricting certain types of access through the submission interface (such as CAPTCHAs) or through usage limits (such as post or tagging quota). The `nofollow` HTML attribute of hyperlinks can also serve as a spam deterrent, since it instructs search engines that a hyperlink should not influence the link target's ranking in the search engine's index, thereby removing the main motivation of spammers.

Demotion-based strategies focus on reducing the prominence and visibility of content likely to be spam. Rank-based methods, for instance, try to produce orderings of the system's content that are more accurate and more resistant to spam [5]. A demotion-based strategy for combating spam is described by [5] and described in more detail in [8]. They construct a simplified model of tagging behavior in a social bookmarking system, and compare different ranking methods for tag-based browsing. They investigate the influence of various factors on these rankings, such as the proportion and behavior of spam users and tagging quota [8].

Spam detection methods, finally, are used to identify likely spam either manually or automatically, and then act upon this identification by either deleting the spam content or visibly flagging it as such to the user [5]. To our knowledge, the only published effort of automatic spam detection in the social reference manager context comes from Krause et al. (2008), who investigate the usefulness of different machine learning algorithms and features to automatically identify spam users and their posts[9]. They test their algorithms on a data dump of the BibSonomy system.

Later in 2008, this work on spam detection for BibSonomy was extended by means of the 2008 ECML/PKDD Discovery Challenge workshop[4], which focused on two data mining tasks related to social bookmarking. One of these tasks was detecting spam users in a social bookmarking system. So far, this has been the only TREC-like initiative focusing on the task of spam detection. With a total of 13 submissions, the majority of the participants' approaches used machine learning for the prediction task. Six out of the top eight approaches used a variety of content-based and co-occurrence-based features combined with machine learning algorithms to separate the spammers from the genuine users [6]. One of the top eight challenge submissions used a graph-based algorithm for the detection task. We participated in the challenge with a preliminary version of our approach, described in [1], and finished in fourth position. In this paper, we extend our approach and test it more extensively using other data representations. Furthermore, we use an additional data set based on CiteULike to confirm the general applicability of our method.

Broadening the scope beyond social websites, we can also find a wealth of other anti-spam approaches in related fields such as blogs. Mishne et al. (2005) were among the first to address the problem of spam comments in blogs and used language model disagreement between the blog post itself, the comments, and any pages linked to from the comments to identify possible spam comments [12]. Their work inspired our approach to spam detection in social bookmarking. In 2006, the TREC Blog Track also paid attention the problem of blog spam [13].

[4] http://www.kde.cs.uni-kassel.de/ws/rsdc08/

3. METHODOLOGY

3.1 Task description

One of the two tasks in the 2008 Discovery Challenge was spam detection in a social bookmarking system [6]. We use their definition of the spam detection task to guide our experiments in this paper. The goal of the spam detection task is to learn a model that predicts whether a user is a spammer or not. An added requirement is that the model should be able to accurately classify initial posts made by new users, in order to detect spammers as early as possible. This decision to identify spam in BibSonomy at the user level instead of at the post level implies that all of a spam user's posts are automatically labelled as spam. This decision was justified earlier in Krause et al. (2008) by the observation that users with malicious intent often attempt to hide their motivations with non-spam posts [9]. In addition, Krause et al. also cite workload reduction as a reason for the decision to classify at the user level. In the experiments described in this paper, we use the setup of the Discovery Challenge for our spam detection task and classify spam at the user level in both our BibSonomy and our CiteULike data set, to make for a fair comparison of our results.

3.2 Data Collection

Automatic spam classification approaches typically demand a training or seed set to learn to predict spam characteristics [5], so for us to be able to test our spam detection approach, we needed access to data sets with manually identified spam objects. We were able to gather such spam labels for data sets based on two social bookmarking websites: BibSonomy and CiteULike. The BibSonomy collection came pre-labeled for spam as part of the aforementioned 2008 Discovery Challenge. For CiteULike we annotated a sizable part of the collection ourselves. Table 1 provides statistics for the presence of spam in the CiteULike and BibSonomy collections. The results indicate that spammers tend to add twice as few posts, but two to three times as many tags to their posts on average than genuine users. Tag count therefore seems to be an informative feature for spam prediction; a fact already signaled in [9]. In the next two subsections we go into more detail about how we obtained our spam annotations and about specific characteristics of the two data sets.

3.2.1 BibSonomy

BibSonomy is a system for sharing bookmarks and reference lists of scientific articles. It allows its users to add their academic reference library as well as their favorite bookmarks to their online profile on the BibSonomy website. Articles are stored as their BibTeX representation, including abstracts, and links to the papers at the publishers' websites. Users can also describe their references using tags and use these to browse and discover new and related references.

BibSonomy is used as a testbed for research into various knowledge organizational aspects of social bookmarking by the Knowledge and Data Engineering group of the University of Kassel, Germany. As part of their research efforts, they organized the 2008 ECML/PKDD Discovery Challenge and made a snapshot of their BibSonomy system available in the form of a MySQL dump. This dump consisted of all resources posted to BibSonomy between its inception in 2006 and March 31, 2008. The distinction between bookmarks and BibTeX records is also made in this snapshot. The data set contained flags that identify users as spammers or non-spammers, and these labels were included in the data set for training and tuning parameters. The Discovery Challenge organizers were able to collect data of more than 2,600 active users and more

Table 1: Spam statistics of the BibSonomy and CiteULike data sets. All CiteULike items were treated as scientific articles, since there is no clear-cut distinction between bookmarks and scientific articles on CiteULike. For BibSonomy, these are the counts of the training material combined with the official test set.

	BibSonomy	CiteULike
posts	2,102,509	224,987
bookmarks, spam	1,766,334	
bookmarks, clean	177,546	
articles, spam	292	70,168
articles, clean	158,335	154,819
users	38,920	5,200
spam	36,282	1,475
clean	2,638	3,725
average posts/user	54.0	43.3
spam	48.7	47.6
clean	127.3	41.6
tags	352,542	82,121
spam	310,812	43,751
clean	64,334	45,401
average tags/post	7.9	4.6
spam	8.9	7.7
clean	2.7	3.2

than 36,000 spammers by manually labeling users. This reveals that the BibSonomy data set is strongly skewed towards spam users with almost 14 spam users for each genuine user. Table 1 also shows that spam users in BibSonomy clearly prefer to post bookmarks, whereas legitimate users tend to post more scientific articles.

3.2.2 *CiteULike*

CiteULike is a website that offers a "a free service to help you to store, organise, and share the scholarly papers you are reading". It allows its users to add their academic reference library to their online profile on the CiteULike website. At the time of writing, CiteULike contains around 1,166,891 unique items, annotated by 35,019 users with 245,649 unique tags. Articles can be stored with their metadata (in various formats), abstracts, and links to the papers at the publishers' websites. CiteULike offers daily dumps of their core database[5]. We used the dump of November 2, 2007 as the basis for our experiments. A dump contains all information on which articles were posted by whom, the tags that were used to annotate them, and a time stamp of the post. It does not, however, contain any of the other metadata available in the online service, so we crawled this metadata ourselves from the CiteULike website using the article IDs. After crawling and data clean-up, our collection contained a total of 1,012,898 different posts, where we define a post as a user-item pair in the database, i.e. an item that was added to a CiteULike user profile. These posts comprised 803,521 unique articles posted by 25,375 unique users using 232,937 unique tags.

This self-crawled CiteULike data set did not come with prelabelled spam users or posts as the BibSonomy data set did. We therefore set out to collect our own spam labels for this data set. In this we faced the same choice as the team behind the Discovery Challenge: at which level of the folksonomy should we identify spam usage—users, items, tags, or individual posts? Our CiteULike collection contains over 1 million posts and over 800,000 items, and going through all of these was not practical. Judging all of the more than 232,000 tags was also infeasible, in part because it is simply not possible for many tags to unequivocally classify them as spam or non-spam. For instance, while many spam entries are tagged with the tag `sex`, there are also over 200 valid scientific articles on CiteULike that are tagged with `sex`. We therefore aimed to obtain an estimate of the pervasiveness of spam on CiteULike by identifying spam users. Judging all 25,375 users in the CiteULike data set would still be impractical, so we randomly selected 5,200 users (∼20%) from the data set and asked two annotators to judge these users on whether they were spammers or not. Each user was judged by only a single annotator to save time.

Figure 1 illustrates the straightforward interface we created for the spam annotation process. For each user it randomly selects a maximum of five articles and displays the article title (if available) and the associated tags. It also shows a link to the CiteULike page of the article. Preliminary analysis showed that articles that were clearly spam were usually already removed by CiteULike and returned a *404 Not Found* error. We therefore instructed our judges to check the CiteULike links if a user's spam status was not obvious from the displayed articles. Missing article pages meant users should be marked as spam. In this process, we assumed that although spam users might add real articles to their profile in an attempt to evade detection, real dedicated CiteULike users would never willingly add spam articles to their profile. Finally, we noticed that spam content was injected into CiteULike in many different languages. From the experience of the annotators, most spam was in English, but considerable portions were in Spanish, Swedish, and German. Other languages in which spam content was found were, among others, Dutch, Finnish, Chinese, and Italian.

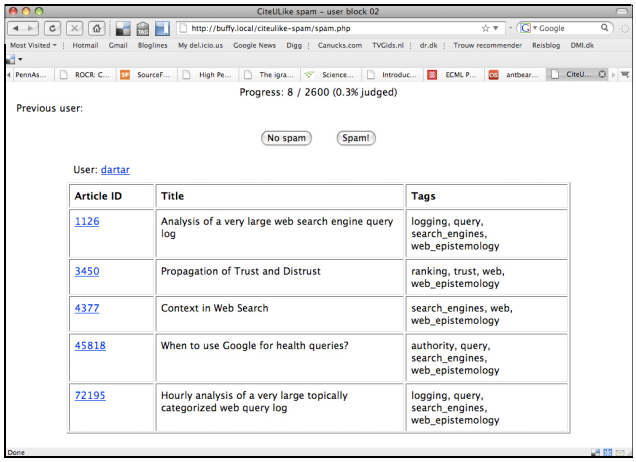


Figure 1: A screenshot of the interface used to annotate a subset of CiteULike users as possible spam users

Of the 5,200 users in our subset, 3,725 (or 28.1%) were spam users, which is a smaller proportion than in BibSonomy. The numbers in Table 1 are reported for this 20% sample of CiteULike users. An extrapolation of these proportions to the full CiteULike data set results in an estimated 7,198 spam users who posted articles to CiteULike. To assess the accuracy of this estimation we may look at the problem from a different angle. As already remarked, certain spam articles are removed quickly from the database by the CiteULike administrators, resulting in *404 Not Found* errors when crawling their article pages. During metadata crawling of all 803,521 articles in our November 7, 2007 data dump, about 26.5% of the articles returned *404 Not Found* errors. A second round of

[5]See http://www.citeulike.org/faq/data.adp.

re-crawling the metadata of these 213,129 missing articles did not change this proportion. While spam removal is not necessarily the only reason for a *404 Not Found* error, we found that 18.7% of the 7,671 users that posted these 213,129 missing articles were spam users identified in our annotation process, which is commensurate with the 20% sample we took. Furthermore, we found that 60,796 of the missing articles (or 28.5%) belonged to the positively identified spam users. These estimates of 7,671 spam users (or 30.2%) and 213,129 spam articles (or 26.5%) strongly suggest that our extrapolation of spam presence on CiteULike is reliable.

3.3 Data Representation

After collecting the data we created a single representation format for all posts, capturing all the relevant metadata in separate fields. As mentioned before, two types of resources can be posted to BibSonomy: bookmarks and BibTeX records, the latter with a magnitude more metadata available. Because there is no such clear distinction in our CiteULike data set, we decided to treat BibTeX records and bookmarks the same and thus use the same format to represent both. We represented all resource metadata in an TREC-style SGML format using 4 fields: `<TITLE>`, `<DESCRIPTION>`, `<TAGS>`, and `<URL>`. URLs were pre-processed before they were used: punctuation was replaced by whitespace and common prefixes and suffixes like `www`, `http://`, and `.com` were removed. Figure 2 shows examples of clean and spam posts in our SGML representation.

A wide variety of metadata fields are available for the posts in the BibSonomy data set. For the bookmarks, the title information is taken from the `book_description` field in the MySQL dump, whereas the `title` field is used for the BibTeX records. The `<DESCRIPTION>` field is filled with the `book_extended` field for bookmarks, whereas the following fields are used for the BibTeX records: `journal`, `booktitle`, `howPublished`, `annote`, `author`, `publisher`, `organization`, `address`, `school`, `series`, `editor`, `description`, `bibtexAbstract`, and `institution`. For both resource types all tags are added to the `<TAGS>` field. URLs, finally, are extracted from the `book_url` and `url` fields, and pre-processed as described above.

Unfortunately, our post representations are significantly poorer for the CiteULike data set: since spam articles are removed from the CiteULike website, we could not crawl the associated metadata of these spam articles (cf. Section 3.2.2). Full metadata is available for the clean articles, but using all metadata of the clean posts and and only the tags of the spam posts would yield an unrealistic comparison. Any classifier would simply learn to predict a post to be spam if it was missing metadata, which is unlikely to be very useful in a real-world situation. We therefore used only the tags for all CiteULike posts, clean and spam alike.

3.4 Evaluation

To evaluate our different approaches and optimized parameters, we divide each data set into a training set, a validation set, and a test set. Our models are trained on the training set, while parameters are optimized on the validation set to prevent overfitting [11]. For the BibSonomy data set, an official test set is supplied as part of the Discovery Challenge as well as training material, so we used this partitioning. We randomly select 80% of the users from the training material for our training set, and assign the remaining 20% to our validation set. This yields a training set of 25,372 users, a validation set of 6,343 users, and a test set of 7,205 users. For the CiteULike data set, we randomly select 60% of all users for our training set, 20% for our validation set, and assign the remaining 20% to our test set. This corresponds to 4,160 training users, 520 validation set users, and 520 users in the CiteULike test set. For the final predictions on the test sets we used only the training sets we created to train our algorithm and generate the spam labeling predictions.

We evaluate our approaches on the validation and test sets using the standard measure of AUC (area under the ROC curve). We optimize k using AUC rather than on measures like accuracy or F-score, as AUC is less sensitive to class skew than F-score [3], knowing that indeed the data is rather skewed, especially in the case of BibSonomy, with 12 spam users to every clean one.

4. SPAM DETECTION

4.1 Language Modeling for Spam Detection

Our approach to spam detection is based on the intuitive notion that spam users will use different language as compared to legitimate users when posting content to a social bookmarking system. By comparing the language models of posts made by spammers and posts made by legitimate users, we can use the divergence between the models as a measure of (dis)similarity. After we have identified the k most similar posts or users using language modeling, we classify new users as spam users or genuine users by scoring these new users by how many spam posts and how many clean posts were found to be similar to it.

Language models are a class of stochastic n-gram models, generally used to measure a degree of surprise in encountering a certain new span of text, given a training set of text. The core of most language models is a simple n-gram word prediction kernel that, based on a context of two or three previous words, generates a probability distribution of the next words to come. Strong agreement between the expected probabilities and actually occurring words (expressed in perplexity scores or divergence metrics) can be taken as indications that the new text comes from the same source as the original training text. Language models are an essential component in speech recognition [7] and statistical machine translation [2], and are also an important model in information retrieval [14]. In the latter context, which we also adopt here, separate language models are built for each document, and finding related documents to queries is transformed into ranking documents by the likelihood, estimated through their language model, that each of them generated the query.

In generating our document language models, we have a range of options on the granularity level of what span of text to consider a document. At the most detailed level, we can construct a language model for each individual post, match these to the incoming posts, and use the known spam status of the k best-matching posts already in the system to generate a prediction for the incoming posts. We can also take a higher-level perspective and collate all of a user's posts together to form merged documents that could be considered "user profiles", and generate language models of these individual user profiles. Incoming posts or users can then be matched against the language models of spammers and clean users to classify them as being more similar to one or the other category.

Figure 3 illustrates these two approaches. In the user-level approach depicted in Figure 3(a), the new user's profile—the merged collection of posts made by this user to the system—are matched against all existing profiles. The most similar users then determine the spam label. In the post-level approach in Figure 3(b), each of the new user's posts is matched against all the posts in the collection. The best matching posts help determine the final spam label of the new user.

A third option—at an even higher level of granularity—would be to only consider two language models: one of all spam posts

<DOC>
<DOCNO> 694792 </DOCNO>
<TITLE>
When Can We Call a System Self-Organizing
</TITLE>
<DESCRIPTION>
ECAL Carlos Gershenson and Francis Heylighen
</DESCRIPTION>
<TAGS>
search agents ir todo
</TAGS>
<URL>
springerlink metapress openurl asp genre article issn
0302 9743 volume 2801 spage 606
</URL>
</DOC>

(a) Clean post

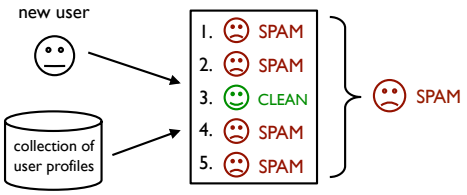
<DOC>
<DOCNO> 2775810 </DOCNO>
<TITLE>
How To Build Traffic To Your Blog
</TITLE>
<DESCRIPTION>
-
</DESCRIPTION>
<TAGS>
blogging directory promotion traffic
</TAGS>
<URL>
webpronews ebusiness sitepromotion
20041210HowToBuildTrafficToYourBlog
</URL>
</DOC>

(b) Spam post

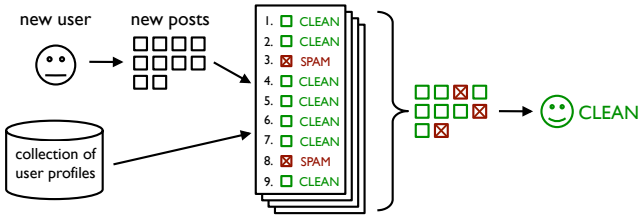
Figure 2: Examples of clean and spam posts in our SGML representation

and one of all clean posts. We believe this to be too coarse-grained for accurate prediction, so we did not pursue this further. Another extension to our approach could have been to use language models for the Web pages or documents that the posts link to, as proposed by[12]. However, it is far from trivial to obtain the full text of all the source documents linked to by the BibSonomy and CiteULike posts. Furthermore, we suspect that incorporating language models from all externally linked Web pages and documents would slow down a real-time spam filtering system to an undesirable degree.

We used the Kullback-Leibler divergence metric to measure the similarity between the language models. KL-divergence measures the difference between two probability distributions Θ_1, Θ_2 according to:



(a) User-level spam detection



(b) Post-level spam detection

Figure 3: Two levels of spam detection approaches

$$KL(\Theta_1||\Theta_2) = \sum_w p(w|\Theta_1) \log \frac{p(w|\Theta_1)}{p(w|\Theta_2)} \quad (1)$$

where $p(w|\Theta_1)$ is the probability of observing the word w according to the model Θ_1 [11, 12].

The Indri toolkit[6] implements different retrieval methods based on language modeling. We used this toolkit to perform our experiments and construct and compare the language models of the posts and user profiles. The language models we used are maximum-likelihood estimates of the unigram occurrence probabilities. We used Jelinek-Mercer smoothing to smooth our language models, which interpolates the language model of a post or user profile with the language model of a background corpus; in our case this is the training collection of posts or user profiles. We chose Jelinek-Mercer smoothing as it has been shown to work better for verbose queries than other smoothing methods such as Dirichlet smoothing [16]. Preliminary experiments with Dirichlet smoothing also showed this to be true for our approach, as it was consistently outperformed by Jelinek-Mercer smoothing.

While it is certainly possible to use other measures of document similarity, such as the cosine similarity, preliminary experiments with the Vector Space model and the cosine similarity metric consistently underperformed the language modeling approach. We therefore focused our efforts on language modeling and KL-divergence for user-user and post-post similarity.

We experimented with both the user-level approach and the post-level approach as illustrated in Figure 3. At the user level, we compared the language models of the user profiles in our validation and test sets with the language models of the profiles in our training set. We then obtained a ranked list of the best-matching training users for each test user. We did the same at the post level by comparing the test post language models with the language models of the training posts. Here, ranked lists of best-matching posts were obtained for each test post. These similarity rankings based on the original similarity scores $sim_{original}$ were normalized into [0, 1]

[6]Available at http://www.lemurproject.org

using the maximum and minimum similarity scores sim_{max} and sim_{min} using the formula from Lee (1997):

$$sim_{norm} = \frac{sim_{original} - sim_{min}}{sim_{max} - sim_{min}} \quad (2)$$

In our BibSonomy data set we have four different metadata fields available to generate the language models of the posts and user profiles in our training collection: title, description, tags, and tokenized URL. In addition to these 'complete' runs with all fields, we also ran experiments where we only used the information from the four fields separately. An example would be to use only the tags from the training users and the test users. This resulted in five different runs for BibSonomy. For CiteULike we only had the tags available, so we performed only one run here. Finally, another option we tried was using all of the available metadata fields in the training set, but restricting the information used of the users and posts in the validation and test sets. This resulted in four extra runs on the BibSonomy data set, one for each metadata field.

4.2 Spam Classification

After we generated the language models for all posts and user profiles, we obtained the normalized rankings of all training documents relative to each test post or user profile. For each of the best-matching training documents, we used the manually assigned spam labels to generate a single spam score for the new user. The simplest method of calculating such a score would be to output the spam label of the top-matching document. A more elegant option would be to take the most common spam label among the top k hits. We settled on calculating a weighted average of the similarity scores multiplied by the spam labels, as preliminary experiments showed this to outperform the other options. Spam scores $score(u_i)$ for a user u_i were calculated according to the following equation:

$$score(u_i) = \frac{\sum_{r=1, r \neq i}^{k} sim(u_i, u_r) \cdot label(u_r)}{k} \quad (3)$$

where for the top k matching users u_r from ranks 1 to k the similarity score $sim(u_i, u_r)$ between the user in question u_i and the matching user u_r is multiplied by the spam label $label(u_r)$ of that matching user. The total weighted scores are divided by the number of matches k, yielding a weighted average score for u_i.

For post-level classification, this meant we obtained these weighted average spam scores on a per-incoming-post basis. To arrive at user-level spam scores, we then matched each incoming post to a user and calculated the average per-post score for each user. Equation 3 was then used to calculate the final spam scores. In the rare case that no matching documents could be retrieved, we resorted to assigning a default label of no spam ('0'). Our default classification was to predict a clean user, as for BibSonomy, for instance, these 0.7% of test users for which no matching documents could be retrieved were legitimate users in 84.2% of the cases.

To make the final classification step, the remaining question is how many of the top matching results should be used to predict the spam score. In this, our approach is similar to a k-nearest neighbor classifier, where the number of best-matching neighbors k determines the prediction quality. Using too many neighbors might smooth the pool from which to draw the predictions too much in the direction of the majority class, while not considering enough neighbors might result in basing too many decisions on accidental similarities. We optimized the optimal value for k for all of the variants separately on the AUC scores on the validation set. These optimal values of k were then used to calculate the final scores on the test sets.

5. RESULTS

Table 2 lists the outcomes of our different spam detection approaches on the two collections. Since we optimized on the validation sets, we mainly focus on the test set scores to draw our conclusions. The best performing approach on BibSonomy, at an AUC score of 0.9661, is spam detection at the user level, using all available metadata fields for both the query and collection posts. The best post-level run on BibSonomy also used all of the data for all of the posts, and achieves a score of 0.9536. On the CiteULike data set, the best performance at the user level and post level yields AUC scores of 0.9240 and 0.9079, respectively. This seems to suggest that our approach generalizes well to other data sets and social bookmarking systems. We observe that in general, using the language models constructed at the user level outperforms using the post-level language models. This is also visible in Figure 4, which shows the ROC curves for the best user-level and post-level runs for each collection.

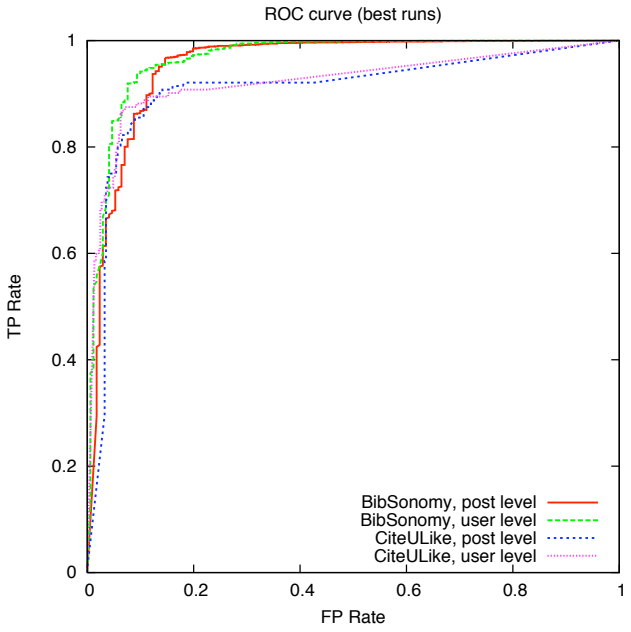


Figure 4: ROC curves of the best-performing user-level and post-level approaches for both collections.

An interesting difference between the validation set and the test set is that using only the tags to construct the language models yields the best performance on the validation set, whereas performance using only tags drops markedly on the test set. Using all available metadata fields results in considerably more stable performance across both BibSonomy evaluation sets, and should therefore be considered the preferred variant.

Another interesting observation is the difference in the optimal size of the neighborhood k used to predict the spam labels. In almost all cases, the post-level approaches require a smaller k than at the user level. The optimal neighborhood size for CiteULike is the same for both the user-level and the post-level approach, and is surprisingly smaller than for BibSonomy.

Finally, comparing the two different sets of BibSonomy runs, using only the matching fields from both the collection and the incoming test posts results in slightly lower scores than when using

Table 2: Results of our approaches on the BibSonomy and CiteULike data sets. Scores reported are AUC, with the best scores for each set of collection runs printed in bold. The two "all fields" rows are one and the same run, but they are repeated here for comparison purposes. The optimal neighborhood size k is listed for each user-level and post-level runs. For the same set of runs, the same value of k was used in both the validation and the test set.

		User level			Post level		
Collection	**Fields**	**Validation**	**Test**	**k**	**Validation**	**Test**	**k**
BibSonomy (matching fields)	all fields	0.9682	**0.9661**	235	0.9571	**0.9536**	50
	title	0.9290	0.9450	150	0.9055	0.9287	45
	description	0.9055	0.9452	100	0.8802	0.9371	100
	tags	**0.9724**	0.9073	110	**0.9614**	0.9088	60
	URL	0.8785	0.8523	35	0.8489	0.8301	8
BibSonomy (single fields in evaluation sets)	all fields	0.9682	**0.9661**	235	0.9571	**0.9536**	50
	title	0.9300	0.9531	140	0.9147	0.9296	50
	description	0.9113	0.9497	90	0.8874	0.9430	75
	tags	**0.9690**	0.9381	65	**0.9686**	0.9251	95
	URL	0.8830	0.8628	15	0.8727	0.8369	15
CiteULike	tags	**0.9329**	**0.9240**	5	**0.9262**	**0.9079**	5

the full data available in the collection and only restricting the fields of the incoming posts.

6. DISCUSSION & CONCLUSIONS

In this paper we presented a adversarial information retrieval approach employing language modeling to detect spam in social reference management websites. We start by using language models to identify the best-matching posts or user profiles for incoming users and posts. We then look at the spam status of those best-matching neighbors, and use them to guide our spam classification. The results indicate that our language modeling approach to spam detection in social bookmarking systems is promising, yielding 0.9536 and 0.9661 AUC scores on spam user detection. This confirms the findings of [12], who applied a similar two-stage process using language modeling to detecting blog spam, albeit on a smaller scale. One particular advantage of our approach is that it could be implemented with limited effort on top of an existing social bookmarking search engine. After any standard retrieval run, the top k matches can then be used to generate the spam classification, requiring only a lookup of predetermined spam labels.

We experimented with using language models at two different levels of granularity, and found that matching at the user level and using all of the available metadata gave the best results. In general, matching at the user level resulted in better performance then matching at the post level for both BibSonomy and CiteULike. This difference can be partly explained by the fact that the spam labels for the users in both data sets were judged and assigned at the user level, as this is the desired level of the end application; even if a spam user posts 'genuine' posts, the entire content of the spam user should be deleted on grounds of the adversarial intentions behind them. Yet, the 'genuine' posts of spam users were automatically flagged as spam, thereby introducing more noise for the post-level classification than for the user-level classification. Early classification of spam users at their earliest posts can therefore be expected to be less accurate than the reported 0.95–0.96 range; post-level AUC scores suggest this accuracy would be closer to 0.91–0.95.

Another likely explanation for the better performance of the user-level approach is sparseness at the post level. A post-level approach is more likely to suffer from incoming posts with sparse or missing metadata. For instance, although 99.95% of all posts in the BibSonomy data set have valid tags[7], this also means that it is possible for incoming posts to have no tags. Without any tags as metadata or sparse metadata in the other fields, our approach cannot find any matching posts in the system. At the user level, this is much less likely to happen: only 0.009% of all users never assign any tags. Aggregating all metadata of a user's posts can yield enough metadata to base reliable predictions on, whereas the post-level approach can be affected by this to a greater extent. Missing tags might also be a reason for the fact that performance on CiteULike is slightly lower than performance on BibSonomy.

In the previous section, we observed that, comparing the two different sets of BibSonomy runs, using only the matching fields from both the collection and the incoming test posts resulted in slightly lower scores than when using the full data available from the collection, and only restricting the fields of the incoming posts. This is probably also a matter of how much data is used: using only matching fields reduces the amount of available metadata for generating the language models, which could make the matching process slightly less effective. We can offer no explanation for the big drop in performance of the tag-based approaches on BibSonomy when comparing the validation set and the test set, other than overfitting on the validation set, as was to be expected.

Finally, when looking at the optimal neighborhood sizes k for BibSonomy, we see that in almost all cases the post-level approaches require a smaller k than at the user level. We believe this is because the presence of multiple topics in user profiles. Individual posts are usually about a single topic, whereas a user profile is composed of all of that user's posts, which are likely to be about multiple topics of interest. This makes finding the related posts to an individual post easier, in the sense that it requires less nearest neighbors to arrive at a prediction. At the user level, however, different parts of a user's profile might match up with different users already in the system, thus requiring more nearest neighbors to arrive at a reliable prediction.

6.1 Comparison with Related Work

With our approach and experimental setup we have improved upon the work described in Mishne et al. (2005) in two ways. One

[7]Valid meaning with a tag other than system:unfiled, the default tag that is assigned by the system when no tags were added by the user.

such improvement in in scale: the data set we used, is several orders of magnitude larger than the one used by Mishne et al., with 43000+ users and 2,300,000+ posts divided over two different data sets, compared to their 50 blog posts and 1024 comments [12]. Since our approach was inspired by theirs, our results serve to confirm that using language model similarity is a good approach for spam detection on a much larger scale as well. Another improvement can be found in the evaluation measure used. In the data set used by Mishne et al. 68% of the content was spam and in our two data sets these proportions were around 93% and 28%. When evaluating on such skewed data sets it is much better to use a measure such as AUC than to use the simple accuracy metric used by Mishne et al. [3].

While we have not compared our approach to standard spam detection approaches such as SVMs or Naive Bayes directly, we can use the outcome of the spam detection task at the 2008 Discovery Challenge to measure our approach against such approaches. Our updated approach would have achieved the third place with an AUC score of 0.9661, surpassed only by a ridge regression approach with extensive feature preparation and selection, and a SVM approach to spam detection [6]. Four of the top eight approaches used SVM; the other three SVM approaches performed worse than the approach described in this paper. One participant used Naive Bayes learning with extensive feature selection, achieving slightly lower scores than our approach. Finally, one of the participants compared five different machine learning methods on the spam detection task, with none of the five outperforming our kNN approach, suggesting that our approach is competitive with other machine learning methods.

7. FUTURE WORK

No spam detection approach can be expected to remain successful without adapting to the changing behavior of the spammers. One way spammers could circumvent our method of spam detection would be by generating metadata with a similar language model to the clean posts in the system. This way, spammers can make it more complicated for our approach to distinguish between them and genuine users. However, this also makes it more difficult for the spammers themselves: it is very hard for a spammer to post resources to a social bookmarking system that will be both similar to existing posts and to the language of the spam entry [12]. In addition, such behavior could easily be countered by extending our method to include the language models of the pages and documents behind the bookmarks. In the case of sparse metadata, this might be able to boost performance of the spam detection algorithm. Extending our approach in such a way is one of the possible avenues for future work. Another option would be to include extra features such as the PageRank scores of the bookmarked pages, and see if pages with low PageRank are more predictive of spam status than others.

The language modeling side of our approach also allows for many different options to be tested, such as symmetric similarity metrics or n-gram occurrence models. For instance, whereas modeling tags can usually be done adequately with unigram models, running text such as the titles and description might benefit from using n-gram models. We leave this for future work; our main goal was to properly validate our spam detection approach on different data sets and at different levels.

Finally, we did not experiment with language models at the class level—one model for all spam content and another for all clean content. While the majority of the related work at the 2008 Discovery Challenge used such a distinction and we compared our approach to these efforts in Section 6.1, it would be interesting to compare such as class-level approach directly with our user-level and post-level approaches. This was not done due to temporal constraints and is also left for future work.

Acknowledgments

We would like to thank Kirstine Wilfred Christensen for her help in annotating the CiteULike data set for spam. The work described in this paper was funded by SenterNovem / the Dutch Ministry of Economics Affairs as part of the IOP-MMI À Propos project, and by the Netherlands Organization for Scientific Research as part of the NWO Vernieuwingsimpuls program.

8. REFERENCES

[1] T. Bogers and A. Van den Bosch. Using Language Models for Spam Detection in Social Bookmarking. In *Proceedings of 2008 ECML/PKDD Discovery Challenge Workshop*, pages 1–12, September 2008.

[2] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. Lafferty, R. L. Mercer, and P. S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, 1990.

[3] T. Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers. *Machine Learning*, 31, 2004.

[4] Z. Gyöngyi and H. Garcia-Molina. Web Spam Taxonomy. In *AIRWeb '05: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, pages 39–47, Chiba, Japan, May 2005.

[5] P. Heymann, G. Koutrika, and H. Garcia-Molina. Fighting Spam on Social Web Sites: A Survey of Approaches and Future Challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.

[6] A. Hotho, D. Benz, R. Jäschke, and B. Krause. Introduction to the 2008 ECML/PKDD Discovery Challenge Workshop. In *Proceedings of 2008 ECML/PKDD Discovery Challenge Workshop*, September 2008.

[7] F. Jelinek. Self-organized Language Modeling for Speech Recognition. *Readings in Speech Recognition*, pages 450–506, 1990.

[8] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating Spam in Tagging Systems. In *AIRWeb '07: Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, pages 57–64, New York, NY, USA, 2007. ACM.

[9] B. Krause, A. Hotho, and G. Stumme. The Anti-Social Tagger - Detecting Spam in Social Bookmarking Systems. In *AIRWeb '08: Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, 2008.

[10] J. H. Lee. Analyses of Multiple Evidence Combination. *SIGIR Forum*, 31(SI):267–276, 1997.

[11] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[12] G. Mishne, D. Carmel, and R. Lempel. Blocking Blog Spam with Language Model Disagreement. In *AIRWeb '05: Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web*, pages 1–6, New York, NY, USA, 2005. ACM.

[13] I. Ounis, M. de Rijke, C. McDonald, G. Mishne, and I. Soboroff. Overview of the TREC 2006 Blog Track. In *TREC 2006 Working Notes*, 2006.

[14] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, New York, NY, 1998. ACM Press.

[15] R. Priedhorsky, J. Chen, S. K. Lam, K. Panciera, L. Terveen, and J. Riedl. Creating, Destroying, and Restoring Value in Wikipedia. In *Proceedings of GROUP '07*, 2007.

[16] C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.

Children's Information Retrieval: how to support children in effective information-seeking?

Hanna Jochmann-Mannak
Human Media Interaction,
University of Twente
PO Box 217, 7500 AE Enschede.
h.e.mannak@utwente.nl

Theo Huibers
Human Media Interaction,
University of Twente
PO Box 217, 7500 AE Enschede.
t.w.c.huibers@utwente.nl

Ted Sanders
Utrecht Institute of Linguistics
UiL-OTS, Utrecht University
Trans 10, 3512 JK Utrecht.
ted.sanders@let.uu.nl

ABSTRACT
In this paper, we describe the objectives of our research project in which we aim to design a search interface in ways consistent with children's needs, cognitive development and thinking style to support children in effective information-seeking.

Categories and Subject Descriptors
D.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *query formulation, retrieval models, search process.*

General Terms
Design, Experimentation, Human Factors.

Keywords
Children, search behaviour, search interfaces.

1. INTRODUCTION
Digital media play an important role in young people's lives. Every day, more children have access to the internet. Even young children already use the internet for playing games or learning. Children seem to manage quite well in working with digital media and searching for information on the internet. But do they really find relevant information as easily as we might think? Do search interfaces support children in effective information-seeking?

2. CHILDREN'S INFORMATION RETRIEVAL PARADIGM
The domain of children's information retrieval (IR) is not limited to searching or browsing on search interfaces. Besides searching for information on an interface, the following components of the search process are also important. The child (1) must have some kind of information need, (2) has to conceptualize this need in his mind, (3) has to translate this need in a question (a search query) and finally, (4) the child has to present this query to an information system. After that, an information retrieval system (5) has to process this query, (6) has to match this query with the information world and (7) has to score, rank and present relevant results. Finally, (8) the child has to understand these results and select a relevant result to satisfy his information need.

In general, these components of the IR paradigm are the same as for adults, but can we really use the same paradigm for children and for adults? Or are there differences between children's and adults' information-seeking behaviour? If that is the case, we need to work towards filling in an IR paradigm especially for children. We need to know what children think and what choices they make in the process from information need to query formulation. Only in that way we can effectively support this process for children. In this section, we describe what previous research already taught us about children's information-seeking. Is there a difference in the process from information need to query formulation between children and adults and what are these differences then?

2.1 Children's search behaviour
In searching on the web, the two most important search strategies are searching and browsing. Keyword searching relies on recall; and browsing relies on recognition. A general assumption is made by researchers that browsing-oriented search tools are better suited to the abilities and skills of children than are keyword search tools (Borgman et al., 1995), because recognition imposes less cognitive load than recall. However, the difference in performance on these different search tools depends on all kind of factors such as the level of abstraction of offered terms (concrete vs. abstract), the kind of search task (open vs. closed), or the extent in which the search tool supports the child in formulating a query. That is why research on searching versus browsing shows different results (Borgman et al., 1995; Hutchinson et al., 2006; Druin, 2003; Bilal, 2000; Schacter et al., 1998).

Research on the differences between children's and adults search strategies (Bilal and Kirby, 2002), showed that children were more chaotic in their search performance than adults. They made more web moves, looped searches and hyperlinks more often, backtracked more often and deviated more often from their target. The researchers concluded that adults adopt a linear or systematic browsing style whereas most children had a chaotic 'loopy' style.

2.2 Information-seeking problems
Most problems with children's information-seeking are due to the fact that search interfaces are designed by adults and based on adult's experiences and preferences. This causes problems for

The 9th Dutch-Belgian Information Retrieval Workshop (DIR 2009), February 2-3, 2009, Enschede, The Netherlands.

children, because they have other needs and preferences than adults and their cognitive, social, physical and emotional development has not yet reached the adolescent formal operational stage of development (Piaget and Inhelder, 1969, in Cooper, 2005). First of all, this causes problems concerning motor skills, because children have difficulties using a mouse. Also typing is difficult for them and takes much time, because they constantly have to search for the right key.

Formulating a search query is also difficult for children, because they have less knowledge to base recall on than adults (Borgman, et al., 1995; Hutchinson, 2006). They tend to use natural language, that makes deciding on a single keyword difficult. For keyword searching, correct spelling, spacing and punctuation is needed, but that is also difficult for children. Category search (termed browsing) can also be difficult for children, because they have little domain-knowledge to decide which category is most relevant to their query. Also abstract, top-level headings can cause trouble, because children's vocabulary knowledge is mostly limited to simple, concrete terms (Hutchinson, 2006).

2.3 Results from research on AquaBrowser

We conducted a small experiment with both children and adults on the usability of a particular search interface, called the AquaBrowser, with a word cloud as a term suggestion tool to support children in reformulating their query. In general, we found the same results on children's and adults' search behaviour as in previous research. Most important new finding is that the term suggestion tool did not support children in effective information-seeking, because most suggested terms were to general for the specific queries. Another problem was that the children were distracted by the word cloud. The word cloud was only effective with very open, self imposed search tasks, in which children were open to other related term suggestions.

3. FUTURE RESEARCH

Existing research on children's information-seeking mostly report on navigation style, web moves, search strategies, search performance or search problems. The methods used in these researches such as recording browser activities (Hutchinson, 2006), online monitoring (Borgman, 1995; Druin, 2003), or observation of search sessions, are suitable to test whether particular search interfaces do or do not support children in effective information-seeking. Existing research, as reported in the previous section, showed that there are important differences between children's and adults' search behaviour. That is why we state the urgency of working towards an IR paradigm especially for children. Only on the basis of a children´s IR paradigm can we conduct research on designing search interfaces that are suitable for children.

The process of filling in this paradigm will be rather complex, because we know now that there are differences between children and adults, but we do not know what principles these differences are caused by. Existing research does not provide information about what happens in a child's mind during the process from a particular need to a query.

We think we can provide insight in the fundamental principles underlying children's search behaviour and search strategies and fill in the components of a children´s IR paradigm, by examining the process from information need to query formulation more thoroughly. Therefore, we have formulated four main objectives that will be the focus of our research.

A. What decisions does a child take in formulating a query or deciding on a search strategy given a certain search task?
B. What influence does the kind of information need (search task) have on the process from information need to query formulation or to a search strategy?
C. What influence does the context (interface) in which a question is asked (given a certain search task) have on the query formulation or search strategy?
D. When is the process from information need to query formulation good (given a certain search task)?

4. CONCLUSIONS

Children have other needs, skills, search criteria and search strategies than adults. That is why children have difficulties with finding relevant information on search interfaces provided for them on the internet. To our knowledge, existing research still does not provide insight in the fundamental principles underlying children's search behaviour and search strategies. In our research, we will work towards filling in the components of a children's IR paradigm by examining children's process from information need to query formulation. By examining that process, we want to find out how to design search interfaces that are consistent to children's needs, skills and cognitive development to support them in effective information-seeking.

5. ACKNOWLEDGMENTS

Our thanks to The Netherlands Public Library Association (VOB) for funding our research program called The digital youth library.

6. REFERENCES

[1] Bilal, D. 2000. Children's Use of the Yahooligans! Web Search Engine: I. Cognitive, Physical, and Affective Behaviors on Fact-Based Search Tasks. J Am Soc Inf Sci Tec. 51 (2000), 646-665.

[2] Bilal, D. and Kirby, J. 2002. Differences and similarities in information seeking: Children and adults as Web users. Inform Process Manag. 38 (2002), 649-670.

[3] Borgman, C.L., Hirsh, S.G. and Walter, V.A. 1995. Children's Searching Behavior on Browsing and Keyword Online Catalogs: The Science Library Catalog Project. J Am Soc Inf Sci Tec. 46 (1995), 663-684.

[4] Cooper, L.Z. 2005. Developmentally Appropriate Digital Environments for Young Children. Library Trends. 54 (2005), 286-302.

[5] Druin, A. 2003. What Children Can Teach Us: Developing Digital Libraries for Children. Library Quarterly. 75, 20-41.

[6] Hutchinson, H., Bederson, B.B. and Druin, A. 2006. The Evolution of the International Children's Digital Library Searching and Browsing Interface. In Proceedings of the IDC (Tampere, Finland, 7-9 June, 2006). 105-112.

[7] Schacter, J., Chung, G.K.W.K. and Dorr, A. 1998. Children's Internet Searching on Complex Problems: Performance and Process Analyses. J Am Soc Inf Sci Tec. 49, 840-8.

Programming by Clicking

[Building Web Robots for Non-Programmers]

Wouter Roelofs
Institute for Computing and Information Science
Radboud University Nijmegen
P.O. Box 9010, 6525 ED
Nijmegen, The Netherlands
wroelofs@student.ru.nl

Alessandro Tadeo Paula
HyperLeap
P.O. Box 1308, 6501 BH
Nijmegen, The Netherlands
aldo@hyperleap.nl

Franc Grootjen
Donders Centre for Cognition
Radboud University Nijmegen
P.O. Box 9104, 6500 HE
Nijmegen, The Netherlands
grootjen@acm.org

ABSTRACT

When deployed into a business setting, web crawlers often require technical skills and education to be built and maintained. This paper presents techniques which can be used to implement a tool with an intuitive graphical user interface that enables unskilled users to generate web robots just by clicking.

Categories and Subject Descriptors

H.3.1 [**Information Storage And Retrieval**]: Content Analysis and Indexing; H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Web Crawler, Subtree Matching, Levenshtein Matching

1. INTRODUCTION

Automatic unsupervised gathering of semi structured web content is not a trivial task. Web pages are mostly designed for human readers: they contain a lot of detail (for example layout) which complicates the collecting process. Moreover web pages are frequently subject to changes. Even if these changes do not affect the desired information, they often break elementary static harvesting techniques.

Confronted with these problems, most current solutions to automatic gathering of web content involve the construction of crawling software. These programs (usually written in script languages) are website specific, have an high maintenance rate, are error prone and hard to debug. Obviously the construction of these crawlers requires skilled (and costly) programmers.

HyperLeap [1] is an independent content provider on the field of infotainment[1]. In order to provide a correct, up to date, consistent and comprehensive overview, HyperLeap gathers information from a large number of sources. To facilitate the gathering process, HyperLeap developed a tool called HyberBot which enables non-programmers to create web robots by clicking.

[1]Infotainment = Information + Entertainment

2. PROGRAMMING BY CLICKING

The rationale behind HyperBot is simple: instead of formally describing (or programming) a web robot, we simply provide examples and let HyperBot infer our intentions and generate the program code.

The programming session starts with visiting the web page of interest. The example data presented to HyperBot consists of two major components:

1. A *path* to the node in the web page that contains the information. The web site's source can be represented by a tree structure. All elements of the web page visible in a browser are located somewhere in this tree and have a unique path that describes their location.
2. The *selection* inside this node that specifies the exact information. It is possible that a node contains more information than the string we want to extract. For example, a node may contain the string 'vr 16 jan | 12:30 uur | Doornroosje Grote zaal', while the web robot wants to extract the timestamp '12:30'.

We now briefly discuss how the example data will be used to generate a web robot.

2.1 Non-Strict Subtree Matching

As already discussed, web sites change frequently. In order to cope with these changes, it is necessary for the web robot to identify the nature of the change. When the change occurs somewhere in the document's tree outside the path that leads to the relevant data, no action is needed. However, if a change affects the path of the relevant data, the node cannot be located by simply following the path. In these cases, we use subtree matching.

Consider a path to be a string formed by several substrings separated by a '/'. So a path p can be written as $p = p_1/p_2/\ldots/p_n$ (n ≥ 1). These substrings p_i are called *levels*.

Obviously, a path is a tree for which each node has at most one child. So locating the node of interest is effectively a subtree match.

A freshly created crawler will always have an accurate path to the relevant node, starting at the root node of the corresponding tree. Subtree matching will always succeed in this case. But when the web site in focus has been structurally altered, matching of the path may fail. In this case the crawler will be less strict and will repeatedly uproot path p (resulting in path $p' = p_2/\ldots/p_n$) until the uprooted path matches or fails entirely. An obvious restriction for uprooting is that there are at least two levels left in p. With each uprooting step, the uncertainty of the result increases. Ongoing research will determine constraints for the uprooting process. Possible indicators include:

- A maximum number of uprooting steps;
- A minimum number of remaining levels in the path;
- Prohibited uprooting beyond certain critical nodes, e.a. table nodes.

2.2 Levenshtein Matching

Most programmers use regular expressions [2] to select substrings. Although this mechanism is powerful enough to perform most string selection tasks, the used notation is cumbersome and hard to maintain. As an alternative to regular expressions we present a matching algorithm that uses an example string (and its selection).

Assume we are interested in extracting the day of the month from a string. As a descriptive example our algorithm will use the string: '*ma 12 jan*'. We will show that this example string alone is sufficient to extract the day of the month from '*don 3 febr*'.

		m	*a*		*1*	*2*		*j*	*a*	*n*
	0	1	2	3	4	5	6	7	8	9
d	1	2	3	4	5	6	7	8	9	10
o	2	3	4	5	6	7	8	9	10	11
n	3	4	5	6	7	8	9	10	11	10
	4	5	6	5	6	7	8	9	10	11
3	5	6	7	6	7	8	9	10	11	12
	6	7	8	7	8	9	8	9	10	11
f	7	8	9	8	9	10	9	10	11	12
e	8	9	10	9	10	11	10	11	12	13
b	9	10	11	10	11	12	11	12	13	14
r	10	11	12	11	12	13	12	13	14	15

Table 1: Levenshtein matching

The first step of the algorithm is to calculate the Levenshtein distance [3] using the tabular method[2] (see Table 1). The lower right corner will hold the distance between the two strings. Subsequently we derive one of the optimal paths of elementary operations (substitution, insertion and deletion) by starting in the lower right corner, selecting a minimal number in west, north or north-west adjacent cells.

Table 2 shows the localized path together with the coded operations. The path tells us how we can optimally convert the example string into the match string: start by inserting an '*d*', substitute '*m*' by '*o*', substitute '*a*' by '*n*', skip to next letter, delete '*1*' etc. Now notice the shaded operations d_1 s_{23}, which originate from the selection columns of the example string. Applying these operations to the example selection '*12*' results in '*3*'.

1	3	5	5	6	8	8	9	11	13	15
i_d	s_{mo}	s_{an}	b	d_1	s_{23}	b	i_f	s_{je}	s_{ab}	s_{nr}

Table 2: Optimal path with operations

3. CONCLUSION

Two techniques to support automatic unsupervised gathering of semi-structured web content are presented. Using these techniques it is possible to program web robots by clicking.

Experimental results show that programs generated this way are more robust against structural changes compared to traditional handwritten ones, and can therefore effectively be used in a business setting.

4. FURTHER RESEARCH

Although HyperBot is not yet matured, the outcome of this research already shows great promise. However, there are some issues that require more attention. Firstly, we may improve the Levenshtein matching algorithm by supplying multiple example/selection strings. Doing so, it is unsure which resulting path should be used for match string selection. An option would be to pick the one with the smallest Levenshtein distance. Another approach could extend the Levenshtein algorithm to merged graphs to handle multiple example/selection strings. A second issue involves the problems of uprooting which have already been discussed in section 2.1. Finally, since the crawling process is unsupervised, some attention should be paid to validity of the results. A sanitizer might assign a likelihood to each result by using implicit semantic knowledge about the extracted fields. Whenever the likelihood drops below some fixed threshold, manual intervention may be required.

5. ACKNOWLEDGMENTS

This research was supported by SenterNovem as project Belfast.

6. REFERENCES

[1] HyperLeap. `http://www.hyperleap.nl`.

[2] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, Princeton, New Jersey, 1956.

[3] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707–+, Feb 1966.

[2]To stimulate the matching of identical characters we used a weighting scheme in which substitution has penalty 2 while deletion and insertion have penalty 1.

Behind the scenes of the digital museum of information retrieval research

Djoerd Hiemstra[1], Tristan Pothoven[1], Marijn van Vliet[1], and Donna Harman[2]

[2] University of Twente
The Netherlands
{hiemstra, pothovent, vlietwm}@cs.utwente.nl

[1] NIST
Gaithersburg, USA
donna.harman@nist.gov

1. INTRODUCTION

As more and more of the world becomes digital, and documents become easily available over the Internet, we are suddenly able to access all kinds of information. The downside of this however is that information that is not digital becomes less accessed, and is liable to be lost to us and to future generations. Whereas there are many scanning projects underway, such as Google books and the Open Library Alliance, these projects are not going to know about, much less find, the specialized scientific literature within various fields. This short paper describes the beginnings of a project to digitize some of the older literature in the information retrieval field [1]. The paper finishes with some thoughts for future work on making more of our IR literature available for searching.

2. INDEXING AND QUERYING

So far 14 of the older reports, such as Cyril Cleverdon's Cranfield reports and Gerard Salton's ISR reports have been scanned, along with a NIST Monograph containing a IR literature survey from the 1960's, a report on the MEDLARS evaluations, and Karen Sparck Jones's *Information Retrieval Experiment* book.

```
<museum>
(...)
 <item type="book" id="isr-10">
  <title>Document Retrieval System - Optimization and
      Evaluation</title>
  <author>Joseph John Rocchio</author>
  <publisher>Harvard University</publisher>
  <year>1966</year>
  <item id="isr-10-1" type="chapter">
   <title>Preface</title>
   <file source="isr-10/pdfs/frontmatter.pdf"/>
   <fulltext>
    <page nr="1">
     <p>THE COMPUTATION Harvard University LABORATORY
      Cambridge, Massachusetts Scientific Report No.
      ISR-10 INFORMATION STORAGE AND RETRIEVAL</p>
     <p>to The National Science Foundation Cambridge,
      Massachusetts March 1966 Gerard Salton Project
      Director</p>
    </page>
    <page nr="2">
     <p>&#xC2;&#xA9; Copyright, 1965 By
(...)
```

Figure 1: XML document resulting from OCR

The documents were scanned at 600 bpi greyscale and turned into PDF with hidden text via OCR. As the reports are almost all large documents with over 100 pages, scanning typically results in a separate pdf document per book chapter. When the pdfs of a report are added to the museum, its OCR recognized texts are derived (using `pdftotext`), and automatically marked up as shown in Figure 1. The elements *title*, *author*, *publisher*, and *year*, have to be inputed manually when uploading the document.

```
declare function museum:search(
  $query as xs:string, $page as xs:integer) as node()*
{
 let $tquery := tijah:tokenize($query)
 let $nexi := concat ("//item[about(.,", $tquery, ")]")
 let $opt := <TijahOptions returnNumber="{$page*10}"/>
 let $qid := tijah:queryall-id($nexi,$opt)
 let $result := tijah:nodes($qid)
 let $count := ceiling(tijah:resultsize($qid) div 10)
 return <result pages="{$count}"> {
  for $x in subsequence($result, $page*10 - 9, 10)
   let $nexi2 := concat ("//p[about(.,", $tquery, ")]")
   let $opt2 := <TijahOptions returnNumber="1"/>
   let $qr := tijah:query($x, $nexi2, $opt2)
   let $snippet := if ($qr) then $qr else ($x//p)[1]
   return <item id="{$x/@id}" type="{$x/@type}">
          { $x/title, $x/author, $x/file }
            <snippet> { $snippet/text() } </snippet>
          </item>
  }
 </result>
};
```

Figure 2: Example PF/Tijah XQuery

The XML data is inserted in PF/Tijah, an XQuery XML database system called Pathfinder (PF), which is integrated with an XML search system called Tijah [2]. PF/Tijah is developed at the University of Twente in cooperation with CWI Amsterdam and the University of Munich, and can be downloaded as part of the MonetDB/XQuery database system.[1] XML data that is inserted in PF/Tijah can be searched on any granularity. The system does not have the notion of "document": Any element can be retrieved using a keyword query, whether it is an *item*, a *page* or a *p* (paragraph) element. Note that both books and book chapters are tagged as *item* in the data, which allows us to do a simple query for items to retrieve both complete books and

[1] `http://dbappl.cs.utwente.nl/pftijah`

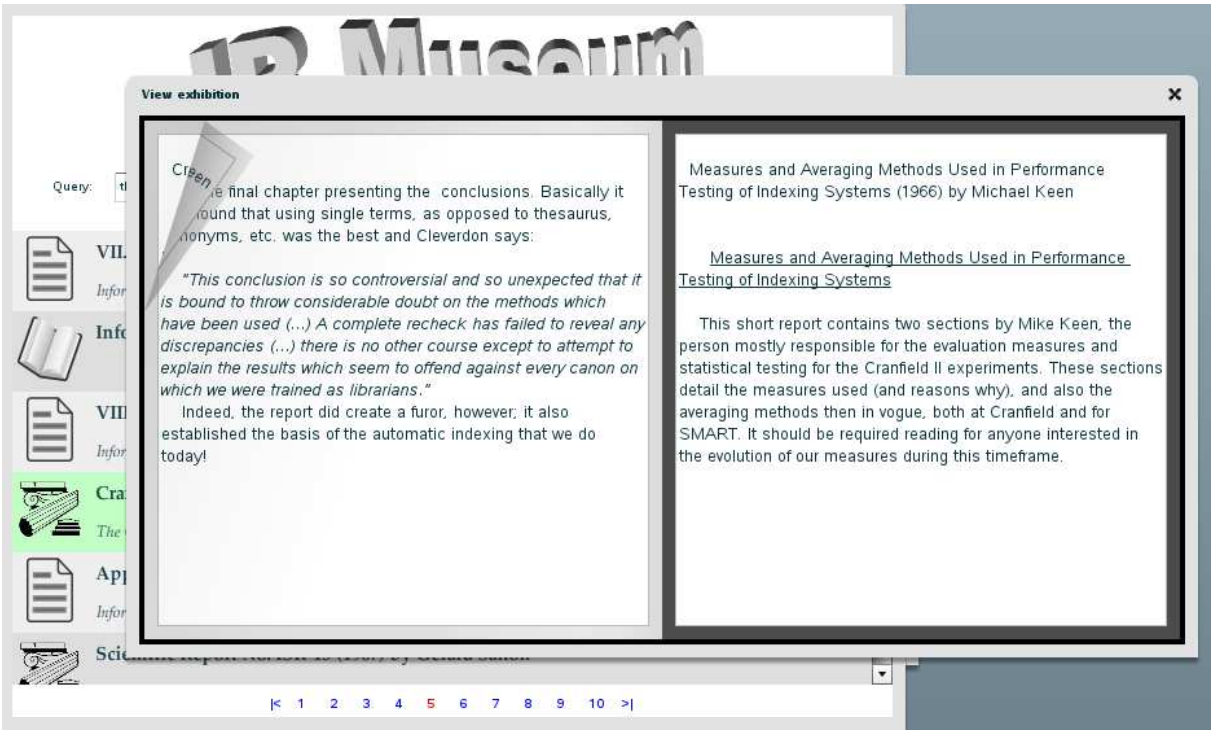


Figure 3: Screen shot showing an exhibition with search results in the back

individual chapters. The example query in Figure 2 shows how PF/Tijah supports powerful searching and result presentation by its query language. The function takes a text query and a result page and searches for *item* elements about the query. It then presents for the 10 best *item* elements its *id*, *type*, *title*, *author*, *file*, and for each element it searches for the best matching paragraph to be presented as a text snippet. All of this is done in one query.

3. EXHIBITIONS

At startup, the museum shows a number of *exhibitions*. The exhibitions contain for every book in the database a small background story. Exhibitions are presented to the user as traditional books with "real" pages that need to be turned by dragging them from right to left in order to go from one story to another. Exhibitions provide a guided tour along the reports and papers that fall under the exhibition's theme. Figure 3 shows a screen shot with a page from the Cranfield exhibition; and the search results partly visible in the back. Users can click directly to the original pdf documents from the exhibitions and from the search results.

All pdfs of the digital museum of information retrieval research are copyright free. They can be made available for download on request. Requests can be directed to the SIGIR by sending an email to `infodir_sigir@acm.org`. The code of the museum itself can be downloaded by choosing "view source" when clicking with the right mouse button.

4. CONCLUSION

What next? Well, we will be continuing the scanning project. The next step is to tackle the reports from the British Library; we are currently compiling a list and contacting them for permission. Several others have offered their books (copyright free) and at some point we would like to seriously go after the Salton books.

In the meantime, please take a look at what we have at: `http://www.sigir.org/museum/`, and learn about who introduced recall and precision, see how the Cranfield and MEDLARS tests were meticulously designed, be amazed by the fact that already in these early papers, full text search outperformed search using manually assigned keywords and thesaurus terms. Also, think of new ways of of accessing the documents. This project can provide some interesting challenges, particularly to the digital library community, including how to better access structured documents, how to deal with vocabulary shift over the years, and studies on how a user community would like to access this type of information.

5. ACKNOWLEDGMENTS

This work was funded in part by the ACM Special Interest Group on Information Retrieval (SIGIR) and by the Dutch Research Program MultimediaN.

6. REFERENCES

[1] Donna Harman and Djoerd Hiemstra. Saving and accessing the old information retrieval literature. *SIGIR Forum*, 42(2):16–21, 2008.

[2] Djoerd Hiemstra, Henning Rode, Roel van Os, and Jan Flokstra. PF/Tijah: Text search in an XML database system. In *Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR)*, pages 12–17, 2006.

Challenges in Professional Search with PHASAR

Cornelis H.A. Koster* kees@cs.ru.nl
Nelleke Oostdijk† n.oostdijk@let.ru.nl
Suzan Verberne† s.verberne@let.ru.nl
Eva d'Hondt† e.dhondt@let.ru.nl

ABSTRACT
The PHASAR (Phrase-based Accurate Search And Retrieval) system is an Information Retrieval and Text Mining system for professional applications. Following the implementation of a prototype in the biomedical domain, we are currently implementing PHASAR for professional search in the intellectual property (IP) domain.

General Terms
Professional search, Interactive search, Intellectual Property

1. INTRODUCTION
Professional search may be distinguished from what could be termed incidental search by the following characteristics: (1) The search is performed by professionals, in their own area of expertise; (2) The search is worth investing some (expensive) time and effort; (3) The search is over a very large collection of documents, many of which may be relevant; (4) The information need is clear but complex, the user can recognize relevant answers; (5) The information need may have to be answered by gathering (passages from) many documents; and (6) Repetitions of the search process with small modifications in the query are routine [3].

The prototype of the PHASAR search system [3] has been developed for professional search on the Medline data collection comprising 18,837,276 scientific abstracts from the biomedical domain. The PHASAR system expects a query to consist of phrases rather than keywords. In an interactive process, the searcher indicates which phrases should occur in the documents for them to be relevant to his/her information need (a form of query-by-example).

Recently, the project *Text Mining for Intellectual Property* (TM4IP)[1] has started at the University of Nijmegen. In this project, the PHASAR system will be implemented for intellectual property search, i.e. search in a database of 9.5 million full-text patent documents. It has been observed that patent searchers prefer Boolean search over ranked search because they desire full control over precision and recall. They are willing to invest work in order to ensure that they retrieve all relevant information pertaining to a query. These user characteristics match well with PHASAR's interactive formulation of phrase queries.

In this poster, we present the PHASAR search system in its current form and we discuss the challenges that we meet in implementing PHASAR for the intellectual property domain.

2. THE PHASAR SEARCH SYSTEM
In this section, we present the basic principles of the PHASAR system following one specific example: the question "What genes are induced by LPS in diabetic mice?", taken from the set of queries used in the TREC 2007 genomics track[2].

PHASAR performs sentence retrieval and presents the results in the form of short passages with a link to the complete document.

PHASAR uses phrases as terms. In the classical approach, a phrase is a sequence of (consecutive) words (e.g. using the sequence *diabetic mice* as a query instead of the separate words *diabetic* and *mice*). Instead of this type of word sequences, PHASAR uses Dependency Triplets (DTs) as terms. A dependency triplet is a pair of (lemmatized) words with their relation, e.g. `[mouse,ATTR,diabetic]`. PHASAR's DT framework is based on the principle of aboutness. DTs have been used successfully in Question Answering [1] for the precise matching of input questions to their answers. In PHASAR, the DTs are obtained from both the indexed documents and the queries in the following steps: dependency parsing is followed by a transduction to DTs, in which (syntactic) variations are normalized onto a common representation. E.g. PHASAR maps the sentences "TNF-alpha is induced by LPS" and "LPS induces TNF-alpha" to a single representation in the index.

PHASAR expects phrase queries that are matched to

*Dept. of Computer Science, Radboud University Nijmegen
†Center of Language and Speech Technology, Radboud University Nijmegen

[1]See http://www.phasar.cs.ru.nl/TM4IP.html
[2]http://ir.ohsu.edu/genomics/

the index of DTs. In the current PHASAR search interface, the searcher fills (at least two of the three) slots for subject, verb and object. Taking 'query-by-example' literally, we can use the query [LPS] [induce] [TNF-alpha] in order to find passages confirming that "TNF-alpha is induced by LPS". Replacing the contents of one of the slots by a question mark, PHASAR shows (besides the sentences from the corpus that match the query) a list of the terms that occur in the ?-position of the phrase, ordered by document frequency.

The user generalizes and specializes the query interactively. Query generalization can be achieved by either joining multiple terms using the *or* operator, or by using one of the built-in thesauri for selecting a semantic term type. E.g. The semantic type `UMG7-GENE-OR-GENOME:` can be used in the query [LPS] [induce] [UMG7-GENE-OR-GENOME:] in order to find all sentences in which a gene or genome is mentioned to be induced by LPS.
A query can be made more specific, either by adding more terms in the query slots or by setting a context from which the results have to be retrieved. We can e.g. first put the query "`diabetic mice`" and save it as a context, after which we can query [LPS] [induce] [?] in this context to get the answers we search for.

Figure 1 shows a screen shot from the current prototype of the PHASAR search engine[3].

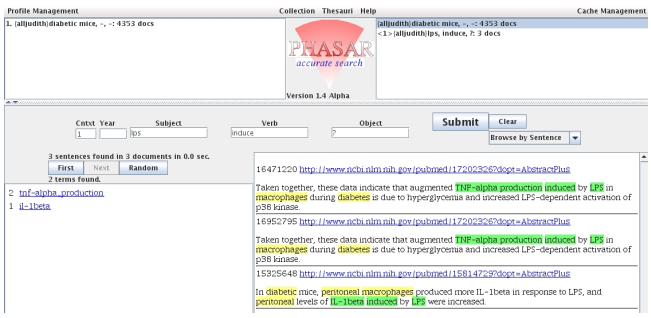


Figure 1: Screen shot of PHASAR displaying the results from the Medline corpus for the structured query [LPS] [induce] [?] in the context of "diabetic mice"

3. CHALLENGES IN TM4IP

In the TM4IP project, the PHASAR system will be implemented for intellectual property data. For this application, we are developing it in two directions: (1) improving the parser and extending the transduction process, and (2) adapting the system to the patent data domain. There are a number of challenges that we will face.

First, in order to improve the accuracy of the dependency parser, it will be turned into a hybrid parser [2] using lexical and triplet probabilities. This requires a bootstrap process that will take time and effort but is expected to lead to much higher accuracy. Patents texts tend to contain very long sentences with many coordinated phrases (see the example below). This a challenge for any parser, which must handle complex coordinations and cope with the ambiguities caused by multiple prepositional attachments.

> A steering system is provided for holding the steering wheel generally parallel to rear wheels on the tractor and for turning the steering wheel through an angle and opposite to a steering angle of the tractor to bring the plow assembly behind the tractor during turns. (doc. `XX000200`)

In the future, we also plan to extend the descriptive framework of the parser grammar with aspects of language that are not directly related to the aboutness of a sentence such as verb modalities and negation.

A second challenge lies in the further extension of the normalization process that takes place in the transduction from parse trees to triplets. In the current version, syntactic variations such as passive versus active voice are already covered (as exemplified in Section 2). A very important addition is the implementation of anaphora resolution using the statistics of the DTs. Using the current parser, we find among the most frequent dependency triplets many triplets with anaphora such as `[it,SUBJ,formed]`. An essential part of the normalization process is to match these anaphora to the correct antecedent, in order to have access to the information contained in the text. Another part of the normalization process is to match synonyms and to resolve abbreviations for technical terms, which are frequent in patents documents.

The last challenge is more of a meta-challenge: evaluating the PHASAR/TM4IP system during its development. We consider two areas of evaluation: (1) evaluation of the accuracy of the hybrid dependency parser and the normalizing transduction, for which we need suitable gold standards, and (2) evaluation of the PHASAR search system on intellectual property data. In 2009 we intend to participate in the first edition of the CLEF-IP track[4]. This will provide us with the opportunity to use common evaluation data. However, since PHASAR expects phrasal queries in an interactive setting, we will not be able to perform a fully automatic evaluation.

4. REFERENCES

[1] G. Bouma, J. Mur, G. van Noord, L. van der Plas, J. Tiedemann, and R. Groningen. Question Answering for Dutch Using Dependency Relations. *LECTURE NOTES IN COMPUTER SCIENCE*, 4022:370, 2006.

[2] K. Foth and W. Menzel. Hybrid Parsing: Using Probabilistic Models as Predictors for a Symbolic Parser. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 44, page 321, 2006.

[3] C. Koster, O. Seibert, and M. Seutter. The PHASAR Search Engine. *LECTURE NOTES IN COMPUTER SCIENCE*, 3999:141, 2006.

[3] Available at http://twoquid.cs.ru.nl/phasar/applet.html

[4] http://www.ir-facility.org/the_irf/clef-ip09-track

Expert Finding of Dutch Politicians

Arjan Nusselder
ISLA, University of Amsterdam
arjannusselder@uva.nl

Maarten Marx
ISLA, University of Amsterdam
KRuislaan 403 1098SJ Amsterdam, The Netherlands
maartenmarx@uva.nl

ABSTRACT

A system is proposed and implemented that creates a language model for each member of the Dutch parliament, based on the official transcripts of the meetings of the Dutch Parliament. Using expert finding techniques, the system allows users to retrieve a ranked list of politicians, based on queries like news messages.

1. INTRODUCTION

Motivation for this research. The Dutch House of Representatives (*Tweede Kamer*) is supplied with information about current and past affairs by its information department, the *Dienst Informatievoorziening* (DI). The DI often pro-actively collects information about topics and events when they suspect one of the politicians will show a special interest in this topic. The DI also performs recommendations of "hot topics" to politicians likely to show an interest in that topic. The DI asked us to implement a system that automates this recommendation process.

Our approach. To match politicians to topics an approach named *expert finding* was used. This approach is detailed in section 4, and based on work by Balog [1]. We used the parliamentary proceedings to build a profile of each politician. A description of the data is given in section 3. The resulting system can be seen as answering the question: "Given the words spoken in parliament by a politician, how well does she match a given text?"

The current approach shows that, given well-structured parliamentary proceedings, it is possible to construct a good performing retrieval system using out-of-the-box information retrieval techniques. Our evaluation using committee descriptions suggests that the current approach has merit and could be explored further, incorporating more advanced techniques.

2. RELATED WORK

The current approach to the retrieval of politicians is based largely on work done by Balog [1]. We used his *Model 1*, which describes the idea of representing experts –politicians in our case– as single documents.[1] This model itself is based on language modelling techniques [3][2].

DIR2009 Enschede

3. DATA

We created a language model of each politician in the Dutch parliament (Tweede Kamer) in the summer of 2008. As textual input data we took the parliamentary proceedings which record everything being said in parliament. Through the PoliticalMashup project [**?**], this data is available in XML in a format which is excellent for our task: every word is annotated with the name of its speaker, her party and the date.

Besides these primary data sources we used biographical data about our politicians available at `www.parlement.com`.

4. METHOD

What needs to be expressed somehow, is the chance that a politician is knowledgeable on –or at least interested in– the topic expressed by a query. To do so, each politician must be represented with a profile. We first define such a profile as a document in which all text related to that policitian is concatenated. This way, the politician–topic matching problem can be reduced to an instance of ranked document retrieval. To calculate the probabilities and ranking, the query is compared to all politicians, each represented as a language model of the concatenation of the related texts.

The measure used for comparison is the Kullback-Leibler divergence. We take $Q : Word \rightarrow Wordcount$ as the function over the words in the query, and $P : Word \rightarrow Wordcount$ as the function over the words in a document representing a politician. The basic formula to calculate the chance of a query given a politician is expressed in equation (1).

$$KL(Q|P) = \sum_i Q(i) \log \frac{Q(i)}{P(i)} \quad (1)$$

The result of a query is a ranked list of document identifiers, corresponding to the politician the texts belong to. To create an accessible and usable interface, the results are embedded in a block of additional information. At the time of writing, an interface is available at `http://zookma.science.uva.nl/politiciansearch/search.php`

For the actual implementation, the Lemur Toolkit was used.[2] The important `Lemur` parameters are *Simple KL* as

[1] See Balog, section 3.2.1.
[2] See: `http://www.lemurproject.org`

6	Commissie voor de Verzoekschriften en de Burgerinitiatieven
8	Financien

Table 1: Names of topics 6 and 8, as they were used as query-text for the evaluation.

Comissie voor de Verzoekschriften en de Burgerinitiatieven Commissie Verzoekschriften en Burgerinitiatieven De commissie voor de Verzoekschriften en de Burgerinitiatieven heeft twee taken: het voorbereiden van een beslissing van de Kamer over een individuele aangelegenheid (waar een burger in een verzoekschrift om heeft gevraagd) en het voorbereiden van een beslissing van de Kamer over de ontvankelijkheid van een burgerinitiatief . . .

Table 2: Beginning of the description of topic 6.

retrieval model, and for smoothing a *Dirichlet prior* set at the total number of word types.

Some additional ideas focussing more on the presentation of the results have been implemented. It is possible to not only collect texts on a per person basis, but also split the aggregations on a temporal or party level. Using a log-likelihood comparison, politicians can then be described as opposed to other politicians, or in a specific time-frame. Extensions like these could improve the usefulness of a system, but are left for future evaluation.

5. EVALUATION

To see how well our approach performs, an experimental evaluation similar to the TREC 2005 W3C enterprise search task was devised.[3] The Dutch parliament has 23 committees, each focussed on a policy topic, roughly corresponding to the existing ministeries[4]. Each committee consists of about eight to twenty-five members, and an equal or smaller number of reserve members. For each committee its name, a short description and its members (all MP's) are known. We used the both the committee names and their descriptions as topics. A result (i.e., a politician) is correct ("relevant") on a topic iff it is an active member of the committee described by that topic (reserve members were not counted). The total number of candidates is 150, which is the number of current members of parliament.

Thus we do two evaluation runs, one with the names of the committees as topics, and one with the descriptions of the committees. Committee names consist of 1 to 5 words (excluding stopwords); descriptions are between 500 and 1000 words. For instance, the description for the finance committee is 638 words (including stopwords).[5] Table 1 gives two examples of committee names; Table 2 contains a part of the description of committee with topic id 6.

These longer descriptions match the purpose of our recommendation system more closely.

Results. We measured the mean average precision (MAP) and precision at 10 (P@10) over two times 23 topics. The results are in Table 3.

Precision at ten is taken as an appropriate measure for two reasons. First, some committees have little more than ten members, which would make precision over ten difficult to evaluate. Second, the intended use of the application foresees a human-readable resultset. Figure 1 shows the P@10 for each topic for both evaluation runs (full description and the committee-name only), with the topics ordered by their P@10 for the description run. Figure 2 additionally shows the MAP score of each topic, ordered by topic id, for the full descriptions topics.

[3]See: `http://trec.nist.gov/`

[4]See: `http://www.tweedekamer.nl/kamerleden/commissies/index.jsp`

[5]The description can be found at `http://www.tweedekamer.nl/kamerleden/commissies/FIN/sub/index.jsp`.

	MAP	P@10
committee names	.38	.48
committee descriptions	.44	.56

Table 3: MAP and P@10 of our experiments.

For the majority of topics –or committees– more than 6 from the first ten results were correct when we used the full description. Looking at figure 1, some possible problems can be identified. Query 8 shows a large discrepancy between the full description and the name only. This may be due to the fact that the topic –just the singe word finance– can be and probably is used in virtually all contexts. The full text of the finance topic is descriptive enough to allow for a match between politicians focused on this area and the committee. The fact that almost all politicians will talk about financial issues however, could make the committee name by itself insufficient. Because the focus of the application lies on a search for more verbose text, this is not necessarily a problem.

Query 6 performs worse both with the full description and only the committee name. Several problems may be the cause of this. First, the committee itself consists –as an exception– of only eight members, which makes it harder to correctly retrieve the correct politicians. Also the topic of the committee is relatively new as compared to others, meaning there is probably less data available to create a profile that acknowledges this specific interest of the members. Third, the topic is pretty vague and seems rather specialized.

6. CONCLUSION

As asked for by the information department of the Dutch parliament, we created a recommendation system which matches current members of parliament to hot topics being described by a piece of text. These are typically news articles. We used an out-of-the-box expert search system based on Model 1 of [1] which showed promising performance using an evaluation similar to that of the TREC 2005 W3C enterprise search task.

A small evaluation (3 topics) which mimics exactly the use-case in mind (finding politicians likely to be interested in a news-story) gave even better results: all topics got a P@10 of .6 or higher. These results can be found at `http://zookma.science.uva.nl/politiciansearch/search.php`. Here the reader can also evaluate the system herself. Interesting queries are "ik" (*I*), "Nederland" (*The Netherlands*) and "vrede" *peace*.

7. REFERENCES

[1] K. Balog. *People Search in the Enterprise.* PhD thesis, University van Amsterdam, September 2008.

[2] D. Hiemstra. *Using Language Models for Information Retrieval.* PhD thesis, University of Twente, 2001.

[3] J. Ponte and W. Croft. A language modelling approach to information retrieval. *Proc. SIGIR '98*, 1998.

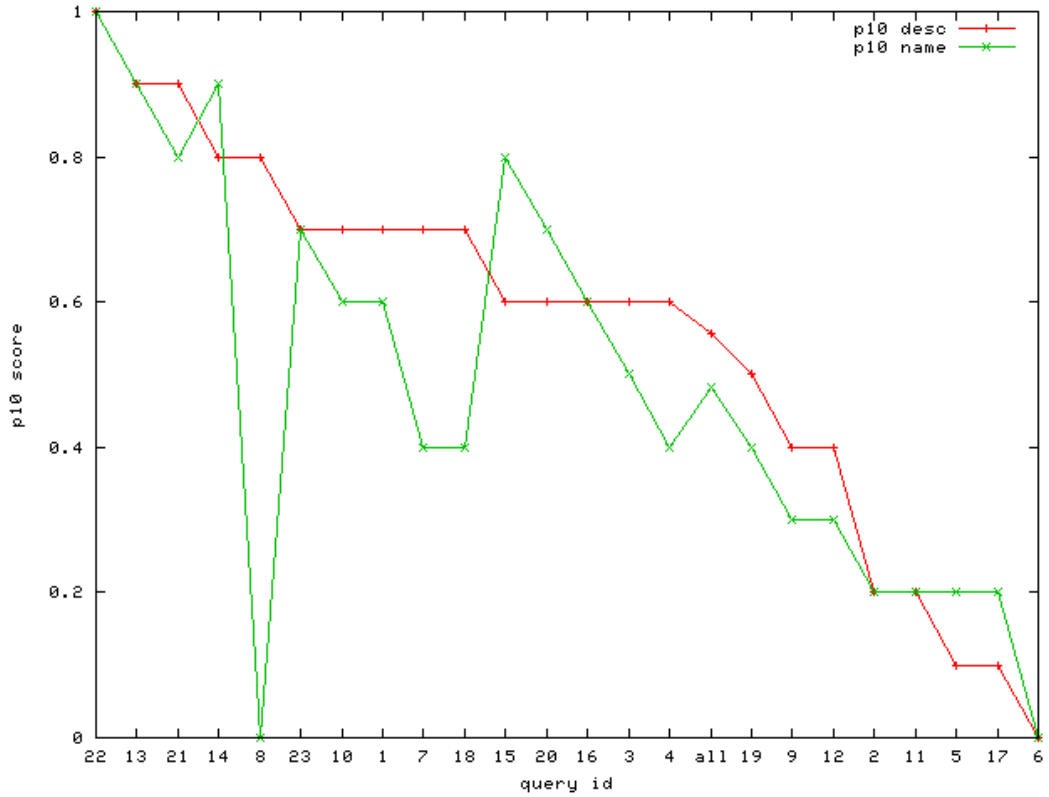


Figure 1: Precision at ten for the full description (desc) and the committee-names (name).

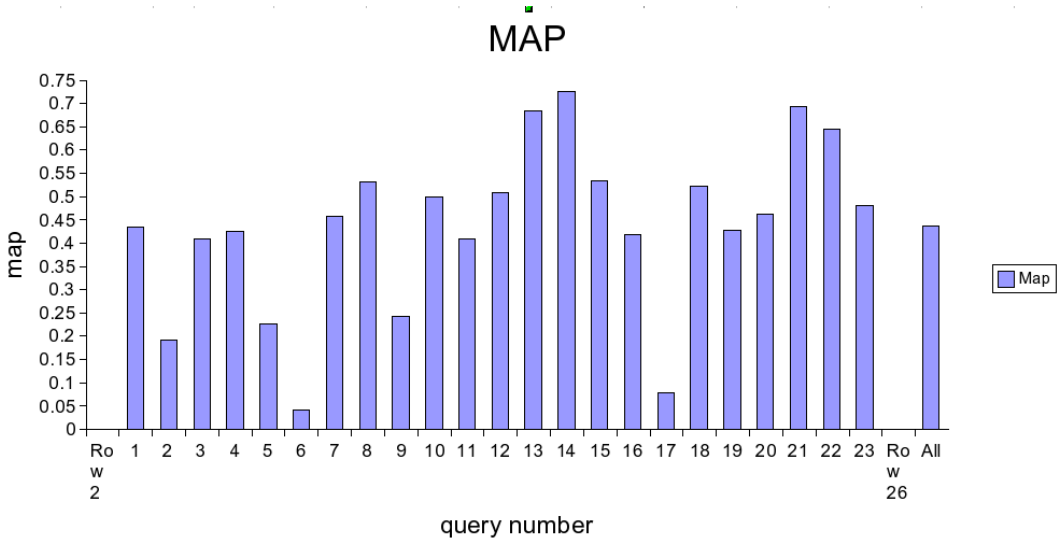


Figure 2: Mean average precision for each full text query.

Previous CTIT workshops

WP 04-01	1st Twente Data Management Workshop, June 2004
WP 05-01	4th Twente Workshop on Cooperative Game Theory, June 2005
WP 05-02	Workshop on Vocabularies, Ontologies and Rules for The Enterprise, September 2005
WP 06-01	2nd Twente Data Management Workshop, June 2006
WP 06-02	European Workshop on Milestones, Models and Mappings for Model-Driven Architecture, July 2006
WP 06-03	Workshop on Integrated Micro Nano Systems, June 2006
WP 07-01	Workshop on Multimodal Output Generation, January 2007
WP 07-02	International Workshop on Social Intelligence Design, & June 2007
WP 07-03	6th Cologne Twente Workshop on Graphs and Combinatorial Optimization, May 2007
WP 07-04	6th International Workshop on Parallel and Distributed Methods in verification, June 2007
WP 07-05	7th International Conference of Computer Ethics, July 2007
WP 07-06	Workshop on Tools for Managing Globally Distributed Software Development, August 2007
WP 07-07	8th International Workshop on Performability Modeling of Computer and Communication Systems, September 2007
WP 07-08	1st international VLDB Workshop on Management of Uncertain Data, September 2007
WP 08-01	2nd International Workshop on Tool Support and Requirements Management in Distributed Projects, August 2008
WP 08-02	Workshops on Management of Uncertain Data \& Quality in Databases, August 2008
WP 08-03	Workshop on Facial and Bodily Expressions for Control and Adaption of Games, September 2008
WP 08-04	Workshop on Models and Model-driven Methods for Enterprise Computing, September 2008
WP 08-05	Workshop on Enterprise Interoperability, September 2008