DIR 2010

10th Dutch-Belgian Information Retrieval Workshop,
January 25, 2010, Nijmegen

SIKS

Werkgemeenschap Informatiewetenschap

nu | STEVIN

NWO
Nederlandse Organisatie voor Wetenschappelijk Onderzoek

_textkernel

Information Foraging Lab

Radboud Universiteit Nijmegen

DIR 2010

Organizing committee

DIR 2010 is organized by the Information Foraging Lab of the Radboud University Nijmegen. The local organization consists of:

- Maarten van der Heijden
- Max Hinne
- Wessel Kraaij
- Maria van Kuppeveld
- Suzan Verberne
- Theo van der Weide

Programme committee

- Toine Bogers (Tilburg University)
- Antal van den Bosch (Tilburg University)
- Gosse Bouma (University of Groningen)
- Peter Bruza (Queensland University of Technology)
- Walter Daelemans (University of Antwerp)
- Martine De Cock (Ghent University)
- Guy De Tré (Ghent University)
- Anne Diekema (Utah State University)
- Djoerd Hiemstra (University of Twente)
- Eduard Hoenkamp (University of Maastricht)
- Veronique Hoste (University College Ghent)
- Theo Huibers (University of Twente)
- Jeannette Janssen (Dalhousie University)
- Jaap Kamps (University of Amsterdam)
- Marie-Francine Moens (University of Leuven)
- Stephan Raaijmakers (TNO-ICT)
- Maarten de Rijke (University of Amsterdam)
- Henning Rode (CWI Amsterdam)
- Tinne Tuytelaars (University of Leuven)
- Remco Veltkamp (Utrecht University)
- Werner Verhelst (Vrije Universiteit Brussel)
- Paul van der Vet (Twente University)
- Arjen de Vries (Delft University of Technology and CWI Amsterdam)
- Jun Wang (University College London)

Table of Contents

Preface

Welcome to the tenth Dutch-Belgian Information Retrieval workshop (DIR) in Nijmegen! Nijmegen is the oldest city in the Netherlands and has a long history of Information Retrieval research. A precursor to DIR was organized in 1996 by Kees Koster, so in fact one could say: DIR is back in Nijmegen. This year DIR is organized by members of the Information Foraging Lab (IFL) - a collaboration between researchers from the Institute of Computing and Information Sciences (iCIS) from the Faculty of Science and the Language and Speech Unit of the Faculty of Arts, both at Radboud University Nijmegen.

The first DIR workshop was organized by Jaap van den Herik in 2000, collocated with the PhD Defense of Ruud van der Pol in Maastricht with Kal Järvelin as keynote speaker. Subsequently, DIR was located in Enschede (2001), Leuven (2002), Amsterdam (2003), Utrecht (2005), Delft (2006), Leuven (2007), Maastricht (2008) and Enschede (2009). The Information Retrieval community in the Netherlands and Belgium has grown significantly over time, judged by the number of accepted papers of Dutch and Belgian origin in top IR conferences and journals.

Information Retrieval also has become a regular component of Dutch Computer Science and Information science programmes. An important driver for this progress has been the involvement in benchmark conferences such as TREC, CLEF and INEX and the growth of the field as a whole due to the web revolution. The change in character and size of the DIR community has also had its impact on the format of DIR itself. Since researchers primarily target high impact conferences and journals, it became increasingly difficult to solicit high quality original work for the DIR workshop, even if workshop papers do not “count” as publications and can be re-published. Therefore, for DIR 2010, we have decided to encourage submissions of “compressed contributions” to present recent original work published by DIR members to the DIR community proper. Judging the number of submissions, this is a success.

We are excited that Elizabeth D. (Liz) Liddy and Cornelis H.A. (Kees) Koster have accepted to be our keynote speakers. Both speakers have a strong interest in the research area that is at the intersection of Information Retrieval and Natural Language Processing.

There were seventeen submissions of original work and eight submissions of already published work under the compressed contributions call. In creating the programme for DIR 2010, the programme committee has given priority to the first category of submissions: The oral sessions consist of seven original papers and one compressed contribution. Seven original papers and seven compressed contributions will be presented in the poster session. As an experiment we returned to a single day workshop format. We hope that a day full of high quality presentations provides a better fit for busy agendas than one and a half workshop day.

We would like to thank SIKS, WGI, Textkernel, NWO and Stevin for their financial support. In addition we thank the Faculty of Science of the Radboud University for hosting the event. Finally, we are thankful to the members of the programme committee for helping to select this excellent programme.

DIR is a community event aimed at making or renewing contacts and fruitful discussions about recent developments in our field. We think that the programme provides ample time for catching up and networking, so we hope you will enjoy this workshop.

Nijmegen, January 2010

Maarten van der Heijden, Max Hinne, Wessel Kraaij, Maria van Kuppeveld, Suzan Verberne and Theo van der Weide

NLP & IR – Wither We Goest?

Elizabeth D. Liddy
Syracuse University
Syracuse, NY, USA
liddy@syr.edu

In the same way that changes in language usage over time are revealing, it is instructive to take both backward and forward glances at how the two disciplines of Information Retrieval and Natural Language Processing have conjoined their individual theories, methods, and technologies to produce understandings and capabilities which had not been imagined earlier but which are either being provided now or imagined for the near future. These bi-directional reviews can provide a rich canvas on which to more pragmatically focus our attention and examine our understandings of the research opportunities and challenges we have at hand and where they might lead.

About Elizabeth D. Liddy

Elizabeth D. Liddy is Dean of the School of Information Studies at Syracuse University, the #1 ranked program for information systems according to U.S. News & World Report. Prior to being named Dean in February, 2008, Liddy was founding director of the Center for Natural Language Processing (CNLP) whose focus is the development of human-like language understanding software capabilities for government, commercial and consumer applications.

Liddy has led 70+ research projects with funding from various government agencies, foundations, and corporate enterprises. She has authored more than 110 research papers and given hundreds of conference presentations on her research. Prior to CNLP, Liddy was the founding president of TextWise LLC, which she led from 1994 to 1999 in the development of an NLP-based search engine which was used by the U.S. Patent Office. In addition, she is inventor or co-inventor on 7 patents in the area of NLP. Liddy has taught in the areas of Natural Language Processing, Information Retrieval, and Data Mining. She is currently Chair of ACM-SIGIR for the 2007-2010 term.

Text Mining for Intellectual Property

Cornelis H. A. Koster
Radboud Universiteit Nijmegen
Nijmegen, The Netherlands
kees@cs.ru.nl

A very important and technically complicated application for Information Retrieval is found in various forms of patent search: the search for "prior art" in a certain technical area and the determination of novelty and patentability of an invention. For these purposes, traditional word-based search engines are not very suitable, due to the complicated linguistic structure and terminology of patent texts. That is why for patent search and biomedical search, somewhat outside the main stream of Information Retrieval, there has been a strong interest in new search techniques, including the use of linguistic phrases as terms instead of keywords, of thesauri and named entity recognition and the aggregation and structured presentation of search results.

In the first part of this presentation, the notion of Text Mining is clarified and contrasted with related notions like Data Mining and Term Extraction. Text Mining typically consists of a Search phase, selecting those documents which relevant to a certain topic, and an Analysis phase, selecting fragments from these documents and resulting in a representation which is suitable for human interpretation.

In the second part of the presentation, the rationale behind the present TM4IP project at Radboud University will be described. The goals of this project are:

- to develop a search engine suitable for text mining, based on deep linguistic techniques (PHASAR)
- to develop an accurate parser suitable for complicated technical English texts (AEGIR)
- to combine these into a Text Mining system for Intellectual Property.

At the end of the presentation, the current version of PHASAR will be demonstrated as an experimental literature search engine accessing 17 million Medline abstracts.

About Cornelis H. A. Koster

Cornelis H.A. (Kees) Koster is emeritus professor of Computer Science at the University of Nijmegen in the Netherlands. Over many years he has done research on Software Engineering, Compiler Construction, Natural Language Processing and Information retrieval. Presently he is leading the Text Mining for IP (TM4IP) project, in which all these interests come together.

A Generative Blog Post Retrieval Model that Uses Query Expansion based on External Collections (Abstract)*

Wouter Weerkamp w.weerkamp@uva.nl
Krisztian Balog k.balog@uva.nl
Maarten de Rijke derijke@uva.nl

ISLA, University of Amsterdam
Science Park 107, 1098 XG Amsterdam

ABSTRACT

To bridge the vocabulary gap between the user's information need and documents in a specific user generated content environment, the blogosphere, we apply a form of query expansion, i.e., adding and reweighing query terms. Since the blogosphere is noisy, query expansion on the collection itself is rarely effective but external, edited collections are more suitable. We propose a generative model for expanding queries using external collections in which dependencies between queries, documents, and expansion documents are explicitly modeled.Results using two external collections (news and Wikipedia) show that external expansion for retrieval of user generated content is effective; besides, conditioning the external collection on the query is very beneficial, and making candidate expansion terms dependent on just the document seems sufficient.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics

General Terms

Algorithms, Theory, Experimentation, Measurement

Keywords

Query modeling, blog post retrieval, external collections, external expansion

1. INTRODUCTION

In the setting of blogs or other types of user generated content, bridging the vocabulary gap between a user's information need and the relevant documents is very challenging. This has several causes: (i) the unusual, creative or unfocused language usage resulting from the lack of top-down rules and editors in the content creation process, and (ii) the (often) limited length of user generated documents. Query expansion, i.e., modifying the query by adding and reweighing terms, is an often used technique to bridge this vocabulary gap. When working with user generated content, expanding a query with terms taken from the very corpus in which one is searching tends to be less effective [6]—topic drift is a frequent phenomenon here. To be able to arrive at a richer representation of the user's information need, various authors have proposed to expand the query against an external corpus, i.e., a corpus different from the target (user generated) corpus from which documents need to be retrieved.

Our aim in this paper is to define and evaluate generative models for expanding queries using external collections. We propose a retrieval framework in which dependencies between queries, documents, and expansion documents are explicitly modeled. We instantiate the framework in multiple ways by making different assumptions.

*The full version of this paper appeared in *ACL 2009*.

DIR'10, January 25, 2010, Nijmegen, the Netherlands.

2. QUERY MODELING APPROACH

We work in the setting of generative language models. Here, one usually assumes that a document's relevance is correlated with query likelihood [4]. The particulars of the language modeling approach have been discussed extensively in the literature and will not be repeated here. Our main interest lies in in obtaining a better estimate of $P(t|\theta_Q)$, the probability of a term given the query model. To this end, we take the query model to be a linear combination of the maximum-likelihood query estimate $P(t|Q)$ and an expanded query model $P(t|\hat{Q})$. We estimate the probability of a term t in the expanded query $\hat{Q}$ using a mixture of collection-specific query expansion models.

$$P(t|\hat{Q}) = \sum_{c \in C} P(c|Q) \cdot \sum_{D \in c} P(t|Q,c,D) \cdot P(D|Q,c). \quad (1)$$

This is our query model for combining evidence from multiple sources. We introduce four instances of the general external expansion model (EEM) we proposed in this section; each of the instances differ in independence assumptions, and estimate $P(t|\hat{Q})$ differently: **EEM1** assumes collection c to be independent of query Q and document D jointly, and document D individually, but keeps the dependence on Q and of t and Q on D.

$$\sum_{c \in C} P(t|c) \cdot P(c|Q) \cdot \sum_{D \in c} P(t,Q|D). \quad (2)$$

EEM2 assumes that term t and collection c are conditionally independent, given document D and query Q; moreover, D and Q are independent given c but the dependence of t and Q on D is kept.

$$\sum_{c \in C} P(c|Q) \cdot \sum_{D \in c} \frac{P(t,Q|D)}{P(Q|D)} \cdot P(D|c). \quad (3)$$

EEM3 assumes that expansion term t and original query Q are independent given document D.

$$\sum_{c \in C} \frac{P(c|Q)}{|\mathcal{R}_c|} \cdot \sum_{D \in \mathcal{R}_c} P(t|D) \cdot P(Q|D). \quad (4)$$

On top of EEM3, **EEM4** makes one more assumption, viz. the dependence of collection c on query Q. Eq. 5 is in fact the "mixture of relevance models" external expansion model proposed by Diaz and Metzler [2].

$$\sum_{c \in C} \frac{P(c)}{|\mathcal{R}_c|} \cdot \sum_{D \in \mathcal{R}_c} P(t|D) \cdot P(Q|D). \quad (5)$$

The fundamental difference between EEM1, EEM2, EEM3 on the one hand and EEM4 on the other is that EEM4 assumes independence between c and Q (thus $P(c|Q)$ is set to $P(c)$). That is, the importance of the external collection is independent of the query. How reasonable is this choice? For context queries such as *cheney hunting* (TREC topic 867) a news collection is likely to offer different (relevant) aspects of the topic, whereas for a concept query such as *jihad* (TREC topic 878) a knowledge source such as Wikipedia seems an appropriate source of terms that capture aspects of the topic. These observations suggest the collection should depend on the query. EEM3 and EEM4 assume that expansion term t and original query Q are independent given document D. This may or may not be too strong an assumption. Models EEM1 and EEM2 also make independence assumptions, but weaker ones.

3. EXPERIMENTAL SETUP

We make use of three collections: (i) a collection of user generated documents (blog posts), (ii) a news collection, and (iii) an online knowledge source. The blog post collection is the TREC Blog06 collection [5], which contains 3.2 million blog posts from 100,000 blogs. Our news collection is the AQUAINT-2 collection, from which we selected news articles that appeared in the period covered by the blog collection (150,000 news articles). Finally, we use a dump of the English Wikipedia from August 2007 as our online knowledge source; this dump contains just over 3.8 million encyclopedia articles. During 2006–2008, the TRECBlog06 collection was used for the blog post retrieval task at the TREC Blog track [5] ("retrieve posts about a given topic") and 150 topics are available.

We report on Mean Average Precision (MAP), precision after 5 and 10 documents retrieved, and Mean Reciprocal Rank (MRR). For determining significance of differences between runs, we use a two-tailed paired T-test and report on significant differences using $^{\triangle}$(and $^{\triangledown}$) for $\alpha = .05$ and $^{\blacktriangle}$(and $^{\blacktriangledown}$) for $\alpha = .01$.

We consider three alternatives for estimating $P(c|Q)$, in terms of (i) query clarity, (ii) coherence and (iii) query-likelihood, using documents in that collection. First, query clarity measures the structure of a set of documents based on the assumption that a small number of topical terms will have unusually large probabilities [1]. Second, the "coherence score" is defined by [3] and it is the fraction of "coherent" pairs of documents in a given set of documents. Third, we compute the conditional probability of the collection using Bayes' theorem. We observe that $P(c|Q) \propto P(Q|c)$ and $P(Q|c)$ is estimated as $P(Q|c) = \frac{1}{|c|} \cdot \sum_{D \in c} P(Q|D)$. Finally, we deploy an oracle approach where optimal settings are obtained by sweeping over them.

4. RESULTS

Results are reported in Table 1. First, our baseline performs well above the median for all three years (2006–2008). Second, in each of its four instances our model for query expansion against external corpora improves over the baseline. Third, we see that it is safe to assume that a term is dependent only on the document from which it is sampled (EEM1 vs. EEM2 vs. EEM3). EEM3 makes the strongest assumptions about terms in this respect, yet it performs best. Fourth, capturing the dependence of the collection on the query helps, as we can see from the significant improvements of the "oracle" runs over their "uniform" counterparts. However, we do not have a good method yet for automatically estimating this dependence, as is clear from the insignificant differences between the runs labeled "coherence," "query clarity," "$P(Q|c)$" and the run labeled "uniform."

model	$P(c\|Q)$	MAP	P5	P10	MRR
Baseline		0.3815	0.6813	0.6760	0.7643
EEM1	uniform	0.3976$^{\blacktriangle}$	0.7213$^{\blacktriangle}$	0.7080$^{\blacktriangle}$	0.7998
	0.8N/0.2W	0.3992	0.7227	0.7107	0.7988
	coherence	0.3976	0.7187	0.7060	0.7976
	query clarity	0.3970	0.7187	0.7093	0.7929
	$P(Q\|c)$	0.3983	0.7267	0.7093	0.7951
	oracle	0.4126$^{\blacktriangle}$	0.7387$^{\triangle}$	0.7320$^{\blacktriangle}$	0.8252$^{\triangle}$
EEM2	uniform	0.3885$^{\blacktriangle}$	0.7053$^{\triangle}$	0.6967$^{\triangle}$	0.7706
	0.9N/0.1W	0.3895	0.7133	0.6953	0.7736
	coherence	0.3890	0.7093	0.7020	0.7740
	query clarity	0.3872	0.7067	0.6953	0.7745
	$P(Q\|c)$	0.3883	0.7107	0.6967	0.7717
	oracle	0.3995$^{\blacktriangle}$	0.7253$^{\blacktriangle}$	0.7167$^{\blacktriangle}$	0.7856
EEM3	uniform	0.4048$^{\blacktriangle}$	0.7187$^{\triangle}$	0.7207$^{\blacktriangle}$	0.8261$^{\blacktriangle}$
	coherence	0.4058	0.7253	0.7187	0.8306
	query clarity	0.4033	0.7253	0.7173	0.8228
	$P(Q\|c)$	0.3998	0.7253	0.7100	0.8133
	oracle	**0.4194**$^{\blacktriangle}$	**0.7493**$^{\blacktriangle}$	**0.7353**$^{\blacktriangle}$	**0.8413**
EEM4	0.5N/0.5W	0.4048$^{\blacktriangle}$	0.7187$^{\triangle}$	0.7207$^{\blacktriangle}$	0.8261$^{\blacktriangle}$

Table 1: Results for all model instances on all topics (i.e., 2006, 2007, and 2008); aN/bW stands for the weights assigned to the news (a) and Wikipedia corpora (b). Significance is tested between (i) each uniform run and the baseline, and (ii) each other setting and its uniform counterpart.

Acknowledgments. This research was supported by the DAESO and DuOMAn project (STE-05-24, STE-0STE-09-12) carried out within the STEVIN programme and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 640.-001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.-004.802.

5. REFERENCES

[1] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIRâĂŹ02*, pages 299–âĂŞ306, 2002.

[2] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2006. ACM.

[3] J. He, M. Larson, and M. de Rijke. Using coherence-based measures to predict query difficulty. In *30th European Conference on Information Retrieval (ECIR 2008)*, page 689âĂŞ694. Springer, Springer, April 2008.

[4] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.

[5] I. Ounis, C. Macdonald, M. de Rijke, G. Mishne, and I. Soboroff. Overview of the TREC 2006 Blog Track. In *The Fifteenth Text Retrieval Conference (TREC 2006)*. NIST, 2007.

[6] W. Weerkamp and M. de Rijke. Looking at things differently: Exploring perspective recall for informal text retrieval. In *8th Dutch-Belgian Information Retrieval Workshop (DIR 2008)*, pages 93–100, 2008.

Comparative Analysis of Balanced Winnow and SVM in Large Scale Patent Categorization

Katrien Beuls
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussels, Belgium
katrien@arti.vub.ac.be

Bernhard Pflugfelder
Matrixware
Operngasse 20b
1040 Vienna, Austria
b.pflugfelder
@matrixware.com

Allan Hanbury
Information Retrieval Facility
Operngasse 20b
1040 Vienna, Austria
a.hanbury@ir-facility.org

ABSTRACT

This study investigates the effect of training different categorization algorithms on a corpus that is significantly larger than those reported in experiments in the literature. By means of machine learning techniques, a collection of 1.2 million patent applications is used to build a classifier that is able to classify documents with varyingly large feature spaces into the International Classification System (IPC) at Subclass level. The two algorithms that are compared are Balanced Winnow and Support Vector Machines (SVMs). Contrary to SVM, Balanced Winnow is frequently applied in today's patent categorization systems. Results show that SVM outperforms Winnow considerably on all four document representations that were tested. While Winnow results on the smallest sub-corpus do not necessarily hold for the full corpus, SVM results are more robust: they show smaller fluctuations in accuracy when smaller or larger feature spaces are used. The parameter tuning that was carried out for both algorithms confirms this result. Although it is necessary to tune SVM experiments to optimize either recall or precision - whereas this can be combined when Winnow is used - effective parameter settings obtained on a small corpus can be used for training a larger corpus.

Categories and Subject Descriptors

H.3.3 [**Information storage and retrieval**]: Information search and retrieval—*clustering, information filtering, retrieval models, search process, selection process*

General Terms

Algorithms, Performance, Experimentation

Keywords

Patent Classification, Intellectual Property, IPC taxonomy

1. INTRODUCTION

The Intellectual Property domain with its more than 70 million patents is characterized by the continuing need to make critical scientific and technical decisions in the face of the exponential growth in the quantity of potentially relevant information. A fundamental need in the Intellectual Property domain is the availability of sophisticated and trustworthy systems for automatic categorization of patent applications.

Various systems have been developed to provide the automatic categorization of patent documents. In general, patent categorization requires the assignment of test documents into a very large taxonomy. In addition, predefined categories are often ambiguous and difficult to distinguish. The processing of patent documents introduces a large number of distinctive terms that are often very technical due to the domain-dependent vocabulary and trigger a low term occurrence frequency over the entire patent collection. The total number of extracted terms to build class profiles is therefore huge. This negatively affects both the categorization quality in terms of Precision and Recall as well as the efficiency of state-of-the-art learning methods.

Text categorization is often defined as the task of assigning a Boolean value to each pair $< d_j, c_i > \in D \times C$, where D is a domain of documents and $C = \{c_1, \ldots, c_{|C|}\}$ is a set of pre-defined categories [13]. Patent document categorization differs from prototypical text categorization on at least three more grounds related to the set of predefined categories. *First*, this set is characterized by a large imbalance as the number of inventions varies in different parts of the taxonomy. *Second*, because the target of the categorization problem is a very small subset of the huge feature space, the scalability of training and testing can become problematic. And *third*, patents are mostly assigned to multiple categories, which means we are dealing with a multi-categorization task.

The Winnow algorithm belongs to the family of on-line mistake-driven learning algorithms such as the Perceptron. It differs from the latter as the algorithm does not learn the linear separation between examples assigned to different categories additively but rather multiplicatively. Most current patent categorization systems, e.g. IPCCAT[1], are based on a variant of the Winnow algorithm, Balanced Winnow.

This paper contrasts Balanced Winnow with an SVM (Support Vector Machine) learner, known to be more robust considering the unbalancedness in class distributions. The computation of the discriminative function of an SVM happens by solving an optimization problem based on a maximum margin separation. SVMs are frequently applied in applications based on images, music, etc. In text categorization, multiple publications [6, 7] state that SVMs outperform other machine learning methods in terms of accuracy but with the disadvantage of needing far more calculation time. The evaluation that is presented in this paper incorporates

[1]Automatic categorization system of the World Intellectual Property Organization (WIPO).

considerably more train and test documents than was the case in previous studies.

Extensive comparisons between algorithms have been published in the literature [13, 4, 15]. This study uses a corpus of 1.2 million patent documents extracted from the Alexandria patent repository of Matrixware[2] and sets up categorization experiments on the basis of the International Patent Classification (IPC) taxonomy on the sub class level (639 classes). The use of such a large corpus makes this study closer to solving the problem such as it exists in the real world. Next to comparing two different learning algorithms, Balanced Winnow and learning with SVMs, the effect of training different patent representations is investigated.

The Linguistic Classification System[3] (LCS), developed in the course of the DORO and PEKING Esprit Projects, was used as a workbench to carry out the experiments. The LCS represents a full text categorization framework with the possibility to adjust various components for optimal performance according to the categorization problem.

2. RELATED WORK

One of the first patent categorization systems was developed by Larkey [9], in which an automatic search and categorization tool for U.S. patent documents according to the US Patent classification (USPC) scheme. The service categorizes patent documents at the subclass level of the USPC by applying the k-nearest neighbor algorithm on a pre-selected set of patent documents retrieved by the system's search component. As the subclasses of those pre-selected patents are known to the system, the k-nearest neighbor algorithm computes a subclass ranking based on the document similarity of the unseen patent and the pre-selection. Unfortunately, an evaluation of the categorization quality was not made public. In 2002, the European Patent Office (EPO) published a comparative analysis of categorization systems in terms of pre-categorization [16]. All compared categorization systems perform pre-categorization on the basis of 44 directorates, 549 teams and 624 subclasses, i.e. three different category structures, with considering only the given prediction. The results show precision scores of 72 %, 57 % and 61 % respectively. The CLAIMS project [2] has participated in a public call of World Intellectual Property Office (WIPO). Again, it does pre-categorization by categorizing patents into the first four hierarchy levels of IPC. While Larkey deployed the k-nearest neighbor algorithm, CLAIMS made an effort to utilize the more accurate Balanced Winnow algorithm using a restricted document representation comprising only the first 600 different terms of every document. In fact, CLAIMS won the public call and the patent categorizer IPCCAT has been directly derived from CLAIMS and is still in use. In [8], another patent categorization system using Balanced Winnow is introduced.

In recent years, international IR evaluation campaigns started patent retrieval and mining tasks. The Japanese evaluation project NTCIR for information retrieval systems was the first IR evaluation campaign which included a separate task on patent mining. A patent categorization subtask [5] was firstly introduced at the NTCIR-5 workshop and categorization (sub) tasks were organised in the following NTCIR campaigns. Most recently, the NTCIR-7 campaign [11] investigated the categorization of research publications into IPC based on training with patent data.

3. CLASSIFICATION METHODS

The categorization of patent documents into the IPC comprises a large number of categories, depending on the level of the IPC at which the categorization will be done. As the standard implementations of Balanced Winnow and SVM only work on binary class problems, ensemble learning is additionally applied by transforming the multi-category problem into an ensemble of binary (one-vs.-rest) problems. For every IPC subclass a separate binary SVM or Balanced Winnow classifier was trained. The categorization is done by selecting the prediction that has the highest confidence among the predictions of all binary classifiers.

3.1 Winnow

The implementation of the algorithm reported in this paper is Balanced Winnow [10, 3]. The classifier consists in this case of weight pairs (positive and negative weights) that are used to calculate the class membership score of a document. The positive weights indicate evidence for class membership whereas negative weights provide negative evidence. The overall weight of a feature is the difference between the positive and negative weights, which are only updated when a mistake is made.

If a mistake is made on a positive example, the positive part of the weight is promoted,while the negative part of the weight is demoted. When a mistake occurs on a negative example the positive part of the weight is demoted and the negative part is promoted. Apart from promotion and demotion parameters α and β on-line algorithms also have a threshold θ that forms the decision criterion for class membership. In Balanced Winnow the thick threshold heuristic is applied. This means that in training, rather than forcing the score of relevant documents above 1 and irrelevant documents below 1 (θ), we have two thresholds: $\theta^+ > 1.0$ and $\theta^- < 1.0$. The result is judged incorrect either if the score of a document is below θ^+ although it belongs to the class or if the document does not belong to the class although its score is above θ^-.

3.2 SVM

In the text categorization process the training data can be separated by at least one hyperplane h'. This presupposes a weight vector $\mathbf{w}^T$ and a threshold b^T, so that all the positive examples are on one side, while the negative examples can be located on the other. This is equivalent to requiring $t_i((\mathbf{w}^T \times \mathbf{x}_n) + b^T) > 0$ for each training example (x_n,t_n). In practice, there can often be several hyperplanes that separate the data but as Support Vector Machines (SVMs) are based on the Structural Risk Minimization principle[4] [14] only the hyperplane that maximizes the margin δ separating positive and negative examples is selected. The small set of training examples that determines the best surface is called the support vectors. They have a distance of exactly δ to the hyperplane.

One problem with the implementation of SVMs is that training fails when the training examples are not linearly separable. Even though this is almost never the case in text

[2] http://www.matrixware.com
[3] http://www.cs.ru.nl/peking/
[4] More information on how SVMs implement structural risk minimization can be found in [7]

categorization, flawless training can result in overfitting of the data and therefore affect the testing accuracy. To avoid this, soft-margin SVM [1] is used. When training with soft margins, an upper bound on training errors is included in the optimization function where this bound is minimized simultaneously with the length of the weight vector. In the SVM implementation SVM Light, the parameter C controls this trade-off between training error and margin. $C = 0$ refers to a hard margin learner, while $C > 0$ represents soft-margin SVM.

4. EXPERIMENTAL SETUP

This section describes the patent data corpus used for the categorization experiments, including some important features of patents. The hardware environment and the setup of the categorization process is also summarized.

4.1 Features of Patent Documents

The content of a patent is governed by legal agreements and is therefore semi-structured. In a European patent document, for instance, the bibliographic data field contains information such as technical details (e.g. the invention title, citations, etc.) and a listing of the parties involved (applications, inventors and agents) but also publication and application references, terms of grant, international convention data and priority claims. A patent's abstract describes in general terms the content of the application whereas the description contains more information on the invention. A more thorough documentation of what has been invented can be found in the description, usually accompanied by multiple tables and figures that support the arguments of the applicant. The claims section of a patent document states the prior art and the novelty of the patent application and often contains standard expressions. A final field that is part of a patent document is the patent's legal status. This status indicates whether the patent is still an application or whether it is an already granted patent.

4.2 International Patent Classification

Patent documents receive specific codes that refer to the class they belong to. The International Patent Classification[5] (IPC) provides a hierarchical system of language independent symbols for the categorization of patents and utility models according to the different areas of technology to which they pertain. In the past, the IPC has been updated every five years and it is currently in the IPC-2009 edition. Each IPC code is a unique combination of the hierarchical structure codes of the patent identity. The three levels in the patent hierarchy that are used in this paper, shown with the number of classes in parentheses, are Section (8), Class (121), and Subclass (631). The official IPC hierarchy contains two deeper levels: Main Group and Sub Group. Table 1 shows a portion of the IPC specification at the start of section G.

4.3 Description of the Corpus

4.3.1 General Statistics

The complete corpus contains 1 270 185 patent documents that are split up into two sub collections: EP (563 248) and PCT (706 937). The patents were extracted from the patent document repository Alexandria Patent created by Matrixware . The patent document archive is provided in a common XML format, created by merging data from various sources. The collection contains EP and PCT patent applications in the period 01/01/1985 - 31/12/2006 having the sections title, abstract and description in English. The patent corpus covers 621 different IPC subclasses, covering 94 % of the possible 639 subclasses according to IPC AL[6]. The focus of research and development varies over the last 20 years and more, resulting in highly varying numbers of patent applications filled for a particular subclasses and year. Consequently, the population of the subclasses with examples is unbalanced in real world and in the data corpus that is illustrated in Figure 1. For instance, the 3 most frequent subclasses in the data corpus are given in Table 2.

Table 1: Sample portion of the IPC taxonomy at the start of Section G

Category	Symbol	Title
Section	G	PHYSICS
Class	G06	COMPUTING; CALCULATING; COUNTING
Subclass	G06N	COMPUTER SYSTEMS BASED ON SPECIFIC COMPUTATIONAL MODELS
Main group	G06N 5/00	Computer systems utilizing knowledge based models
Sub group	G06N 5/02	Knowledge representation

Table 2: The 3 most frequent IPC subclasses in the data corpus

Sub class	Description	Examples
A61K	Preparations for medical, dental, or toilet purposes	121955
G06F	Electric digital data processing	76575
A61P	Therapeutic activity of chemical compounds or medicinal preparations	65655

4.3.2 Sub Collections

Sub collections were created for better indicating both the categorization quality and scalability of the learners on smaller data sets of respectively 100k, 200k, 400k and 500k. A statistics of all sub collections, including coverage of IPC classes and subclasses, is listed in Table 3. Note that the full data set contains double the amount of documents of the biggest sub collection. Train and test sets are built for each of sample collection (cf. 4.4)

Since the class distribution of a collection used for training influences the categorization quality, we defined an algorithm that takes the class distribution of the original corpus into account. Instead of selecting classes that are included into a sub collection randomly, the algorithm follows the following idea. *Firstly*, all classes are ordered based on the number of examples. *Secondly*, k sample sets are created. *Thirdly*, n classes are chosen out of every sample set and all

[5] http://www.wipo.int/classifications/ipc/en/

[6] IPC Advanced Level (AL)

Table 3: Sub collections statistics

Dataset	#Docs	#Classes	#Subclasses
100k	103666	52(40%)	70(11%)
200k	179250	70(54%)	120(19%)
400k	400750	109(84%)	400(63%)
500k	509560	120(93%)	600(94%)
1200k	1270185	121(94%)	631(98%)

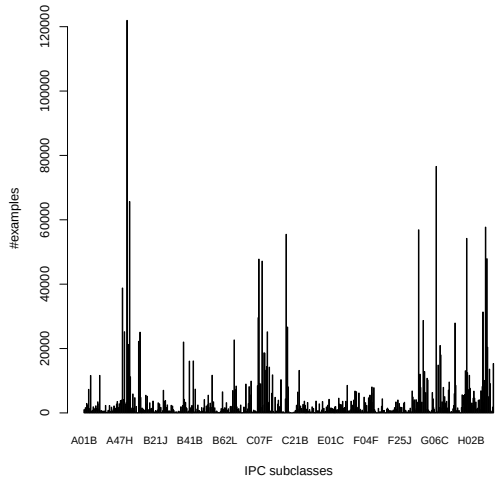


Figure 1: Class distribution of the whole data corpus containing 1.2 million patent applications

examples, which are assigned by those selected classes, are added to the sub collection.

We investigate four different document representations including different sections of the patent documents, shown in Table 4.

4.4 Test Environment and Settings

All the experiments were run on the LDC (Large Data Collider), an SGI Altix 4700 machine provided by the IRF. This server runs under SUSE Linux Enterprise Server 10 and has 40 Dual Core Itanium 2 (1.4 GHz) processing units as well as 320 GB memory. In contrast with the newest version of the LCS, the system that we used for our experiments did not apply parallelization in the training phase.

The experiments were executed using the LCS classification system, using the following setup:

Pre-processing: de-capitalization, removal of special characters such as braces. Term profiles are created by LCS's proprietary indexer. Class profiles containing the term distributions (per class) are generated for the term selection step.

Table 4: Document representations

tabs	title, abstract
itabs	inventor, title, abstract
tdesc	title, description
itabsdesc	inventor, title, abstract, description

Global term selection: document frequency (min=3), term frequency (min=3), noise reduction based on the Uncertainty measure introduced in [12].

Local term selection (class-specific): Simple Chi Square. We chose the LCS feature to automatically select the most adequate number of relevant terms for every class.

Term strength calculation: LTC algorithm [8], extension of TF-IDF.

Training method: Ensemble learning based on one-vs.-rest binary classifiers.

Classification algorithms: Balanced Winnow and SVM, using LCS's propriety implementation of Balanced Winnow and the SVM Light implementation developed by Thorsten Joachims[7].

Parameter settings for the Winnow experiments: A settings of $\alpha = 1.1$, $\beta = 0.9$, $\theta^+ = 1.1$, $\theta^- = 0.9$ and 4 iterations was chosen according to an evaluation carried out within the domain of patent categorization [8].

Parameter settings for the SVM experiments: Based on an evaluation regarding the optimization of F1 using a small subset ($\pm$50000 documents) of the corpus, we selected $C = 1.5$, $J = 1.0$ along with a linear kernel.

Class assignment: min/max number of classifications for each document (resp. 1 and 4)

Evaluation: Split ratio for train/test split: 80/20. The quality of the categorization is determined by Precision, Recall and F1 measure according to the definition in [8]. Both micro-averages and macro-averages are calculated to measure the quality over all classes.

5. EXPERIMENTAL RESULTS

This section presents the experimental results of comparing both learning algorithms SVM and Balanced Winnow on the basis of four document representations defined in Section 5.1. In Section 5.2, the investigation of the SVM parameters complexity (C) and cost factor (J) and the search for an optimal SVM parameter settings concludes this section.

5.1 Comparative Analysis

In Table 5, the micro-averages of Precision, Recall and F1 are listed based on the five sample collections (Table 3) and the four document representations (Table 4). Due to scalability limits, the experiment of SVM with *tdesc* and *itabsdesc* representations on the full data corpus could not be executed, while the corresponding experiment deploying Winnow succeeded. Macro-averages are not presented here due to the space limitations.

Although primary results confirmed the superiority of SVM training over Balanced Winnow with regard to evaluation scores, a detailed analysis of both algorithms should take a complete range of different aspects into account.

First, there is the size of the feature space. On the one hand, the dimensionality of the feature space depends on the document representation that is used in the evaluation. As shown in Table 5, the difference in performance between the

[7] `http://svmlight.joachims.org`

Table 5: Micro-averaged Precision (P), Recall (R) and F1 results for SVM and Balanced Winnow

Doc.Rep.	100k			200k			400k			500k			1200k		
	SVM														
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
tabs	87,97	71,58	78,93	84,94	47,42	58,16	82,38	51,09	63,07	81,01	46,79	59,32	79,21	45,31	57,64
itabs	88,18	70,80	78,54	84,82	48,48	59,00	83,18	52,59	64,44	82,13	48,27	60,81	**80,08**	**47,57**	**59,69**
itabsdesc	**90,40**	79,86	**84,81**	**90,66**	58,54	68,04	**84,47**	**62,26**	**71,68**	**83,00**	**57,48**	**67,92**	x	x	x
tdesc	90,31	**79,90**	84,79	90,51	**76,79**	**83,09**	84,31	61,84	71,35	82,32	56,86	67,26	x	x	x
Doc.Rep.	**Balanced Winnow**														
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
tabs	72,75	77,87	75,23	70.29	74.49	72.33	59,20	57,77	58,48	58,14	53,83	55,90	56,43	52,81	54,56
itabs	73,67	78,57	76,04	71,84	75,07	73,42	62,03	59,68	60,84	60,37	54,85	57,48	59,64	53,94	56,65
itabsdesc	**80,86**	84,30	82,54	**79,66**	**81,99**	**80,81**	**70,30**	**67,27**	**68,75**	**68,22**	**61,57**	**64,72**	**66,37**	**59,59**	**62,80**
tdesc	80,71	**84,67**	**82,64**	79,48	81,82	80,63	67,74	61,46	64,45	67,74	61,46	64,45	65,63	59,49	62,41

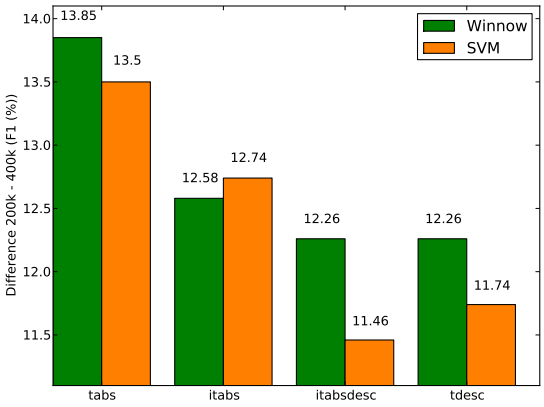


Figure 2: Decrease in F1 scores between 200k and 400k

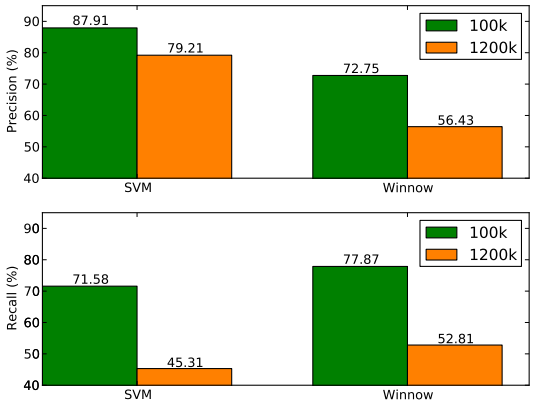


Figure 3: Precision/Recall results on 100k and full corpus for SVM and Winnow (*tabs*)

algorithms becomes smaller with respect to the term space, ranging from the smallest document representation *tabs* to the largest *itabsdesc* and *tdesc* representations. This means that the gap in terms of achieved Precision and F1 between SVM and Winnow narrows when longer documents are used.

On the other hand, the SVM optimization problem seems to be affected more by an increase in the amount of documents that is used. In our experiments, no single documents were added to the corpus but complete classes. The biggest increase in classes occurs when going from the 200k to the 400k collection (70 classes in 200k, 400 in 400k). Different reactions are triggered dependent on the learner. When the feature space is smallest (*tabs*), Winnow is slightly more affected by the increase of classes than SVM. Running from smallest to largest feature space (*itabsdesc*), the SVM bars in Figure 2 decrease more steeply than the Winnow bars. Whereas in Winnow training, the addition of inventors to the document representation has a bigger effect on the adaptation to an increase in classes than the addition of the description, SVM reacts more strongly on the explosion of the feature space in combination with the rapid increase in classes. The difference between 200k and 400k across the learning algorithms is maximized at this point (*itabsdesc*).

Second, both Precision and Recall should be considered when drawing a comparison between both learners. The bar charts in Figure 3 show that both SVM and Winnow expose a less steep decrease in Precision than Recall when moving from the smallest to largest dataset. Whereas Recall drops by 26.27 % for SVM and 25.06 % for Winnow, the difference in SVM Precision is only 8.76 %. Winnow Precision, on the other hand, decreases by double the percentage of SVM Precision, namely 16.32 %. This rather indicates the extremely high Precision of SVM training.

When looking at the micro-averages in greater detail, there are further interesting trends visible. The training results for Winnow show that over the collection sizes, Recall is higher than Precision for 100k whereas the opposite scenario is found for the 1200k corpus. This indicates that Winnow Recall decreases more rapidly than Winnow Precision when more classes are used for training. This is not the case for SVM, where Precision stays always higher than Recall. What is more, SVM Precision scores proved to be so stable that the difference between *tabs* and *itabsdesc* for the 400k collection results in only 2 %, while the exactly same settings yield a difference of 11 % for Balanced Winnow.

Figure 4 illustrates the empirical distributions of category-based Precision and F1 for two different document representations using box plots. The SVM classifier is able to learn most of the sub classes with a quite similar Precision, even though the numbers of examples strongly vary among the sub classes, whereas the Winnow classifier is not that stable. Contrarily, the category-based F1 results vary much stronger in terms of the SVM classifier compared to the Winnow classifier. This gap between Precision and Recall results

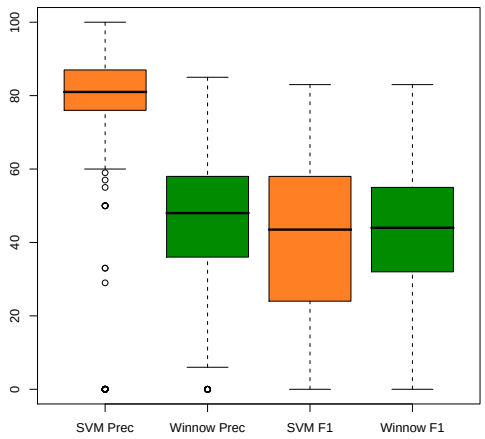


Figure 4: Extreme results of SVM Precision: Boxplots of category-based Precision/F1 results on *itabs* and the 1200k dataset

among the categories suggests the need of parameter tuning in order to balance Precision and Recall.

5.2 Tuning of SVM parameters

So far the experimental results suggest superior performance of the SVM in terms of Precision compared to Winnow. However, the large differences between Precision and Recall for both learning algorithms imply the need of parameter tuning. In fact, the used SVM parameters caused Precision to remain high, while Recall dropped under 50 %. Depending on the goal of the experiment, such a low level of Recall may not be ideal. We therefore target parameter tuning in order to find SVM parameters that optimize either Precision or Recall.

Our previous experiments were conducted with three parameters that affected the workings of the SVM Light package: the soft margin parameter (C), the cost parameter (J) and the type of kernel (T). The latter is kept at its default value, i.e. linear. As the type of problem we are trying to solve is linearly separable, using a polynomial or radial-based kernel would not bring an increase in accuracy but rather delay the categorization even more.

The parameter tuning was carried out with the smallest collection (100k) to speed up the process.

5.2.1 C tuning

The default setting for the soft margin parameter in SVM Light is $avg[x * x]^{-1}$. This parameter setting tries to find the line that best separates the positive and negative training examples. To maximize this difference (x), the value is squared and inverted. This is done for every training example that has a counter example in its neighborhood, i.e. only for support vectors. In general, C can be any floating point number bigger than 0.

Different values of C ranging between 0 and 50 were tested. Above 2.0 the results do not add much more to the parameter tuning (and are therefore not shown): Precision fluctuates in a slightly downward trend from 77.96 % to 76.2 %; Recall finds its maximum at $C = 17.5$ (72.81 %) and fluctuates up and down across the interval (71.38 % at $C = 50$).

A more detailed parameter tuning between 0.1 and 2.0 with steps of 0.1 shows that for the smallest values of C, the distance between Precision and Recall is greatest. The maximum Precision (92.43 %) and minimum Recall (48.06 %) values are situated at $C = 0.1$. The recall curve then rises rapidly between 0.1 and 1 up to 69.78 % (+ 21.72 %). Precision drops only by 3.5 %. The highest recall in this interval is situated at $C = 2.0$ (72.22 %). The C value used in the baseline experiments (1.5) is the point where the F1 value reaches its maximum.

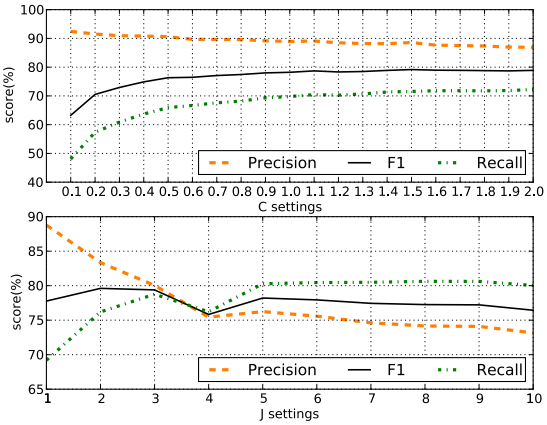


Figure 5: SVM tuning of the complexity parameter C (top) and the cost factor J (bottom) on 100k corpus

5.2.2 J tuning

The second parameter that can be changed in the SVM algorithm used in the experiments is the cost factor. By changing J, the misclassification of a positive example can be punished more or less severely. The cost of misclassifying a positive example is determined as $C_+ = J \times C$, while the misclassification cost of a negative example C_- remains unaltered [6]. By default $J = 1.0$.

Figure 5 shows the results of a tuning $J \in [1, 10]$ with step sizes of 1. The highest recall value is found at $J = 8$. Although the goal of this parameter tuning is to get Recall possibly at its highest level, Precision, and therefore F1, value should in the best cases not drop significantly. Therefore, $J = 5$ seems a reasonable choice. In all these experiments, C was kept at its default value (1.5).

5.2.3 Grid search

In order to find SVM parameters that can be used to optimize Precision and Recall, a grid search of C/J pairs was carried out. Similar to the previous tuning experiments, we define $C \in [0, 2]$ and $J \in [1, 8]$. Figure 6 shows the results of this grid search as heat maps. It can be seen that the extremes in Precision and Recall are almost reversed. Whereas the highest recall values are obtained when J is largest, Precision peaks when J is smallest. This indicates that a separate tuning for Precision and Recall is necessary when SVM is used. Contrarily, a similar investigation of

Table 6: Optimal Precision/Recall settings for SVM and Winnow on *itabsdesc* 100k

Settings	Precision	Recall	F1
$C = 0.25; J = 10$	68.54 %	85.21 %	75.97 %
$C = 0.05; J = 12$	62.46 %	85.55 %	72.20 %
$\alpha = 1.05; \beta = 0.9$	71.43 %	78.73 %	74.90 %
$\alpha = 1.1; \beta = 0.9$	72.23 %	78.00 %	75.00 %

Table 7: Maximum Precision/Recall and default SVM settings on *itabsdesc* 500k

Settings	Precision	Recall	F1
$C = 0.10; J = 1.0$	89.34 %	30.97 %	46.00 %
$C = 0.25; J = 10.0$	62.92 %	73.50 %	66.14 %
$C = 1.50; J = 1.0$	83.00 %	57.48 %	67.92 %

Balanced Winnow tuning[8] identifies one set of parameters that optimize both Precision and Recall. If the C parameter is taken into account, the plots show that the smallest C values exhibit both the highest and lowest Precision and Recall results. Precision peaks at $C = 0.1$ and $J = 1$, whereas Recall reaches its maximum value at $C = 0.4$ and $J = 8$.

The highest Recall scores are situated on the open-ended side of the heat plot. In order to have a quick idea whether we have reached the upper limit in terms of Recall scores, two more experiments were conducted that were expected to improve Recall even more (and therefore worsen Precision). The results, which are summarized in Table 6, indicate that using even higher J values does not improve Recall significantly anymore compared to Figure 6(b).

5.2.4 Robustness

It was verified how the best Precision/Recall settings on the 100k would score on the 500k corpus. Is the accuracy on 500k still proportionally higher with tuning as without? To maximize Precision, the values $C = 0.1$ and $J = 1.0$ were used. The maximum recall was in the 100k tests achieved with $C = 0.25$ and $J = 10.0$. These values were tested on the 500k *itabsdesc* corpus. The results are summarized in Table 7.

The experiment that was run with the optimal Precision settings yielded Precision of 89.34 % on the 500k corpus, which is 6.3 % higher than the baseline run on the 500k *itabsdesc* corpus. On the 100k corpus, the difference between the baseline and the optimal Precision settings was 3.6 %. This shows that the optimal Precision settings hold for a bigger corpus as well. The maximum recall settings yielded an increase in Recall of 16.02 % on top of the baseline. A last point to note is the stability of the F1 value, losing only just over 1 % after the tuning has taken place.

6. DISCUSSION AND CONCLUSIONS

The Support Vector Machine is a popular machine learning algorithm, which achieves excellent performance in many categorization applications, including text categorization. Although patent categorization is a sub problem of text categorization, a direct application of SVM must be evaluated due to specific aspects of patents and patent classification systems. This study investigates SVM in patent categorization based on the IPC and compares SVM with Balanced Winnow, which is frequently applied in current patent categorization systems such as the IPCCAT.

SVM outperforms Balanced Winnow in terms of Precision on all sample collections and document representations (see Table 5). The difference in Precision of the two learning algorithms becomes smaller with growing number of terms. In other words, Balanced Winnow benefits more from the increase in the number of terms introduced by the longer document representations *itabsdesc* and *tdesc*. In fact, SVM still outperforms Balanced Winnow in terms of Precision on longer document representations. On the other hand, in the experiments the SVM training does not scale to longer document representations without algorithmic parallelization and distribution, while training with Balanced Winnow still succeeds.

In contrast to the Precision results, SVM training delivered lower Recall compared to Balanced Winnow. The results show large differences between Precision and Recall for each of the SVM experiments. This suggests that the SVM parameters that were applied in the experiments are not optimal and parameter tuning is needed in order to increase both Precision and Recall.

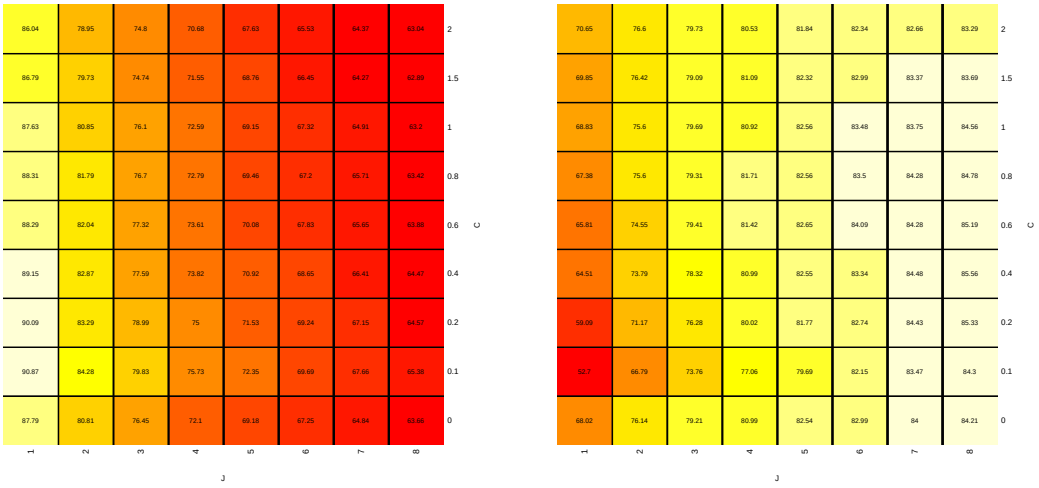
Since the used sample collections do not only grow in terms of document number, but, also in the number of classes (max. 600), the Precision values in Table 5 show that SVM results remain more stable than Winnow results as the corpus size increases (100k $\rightarrow$ 1200k). Another important conclusion is that SVM is more robust regarding the unbalanced class distribution depicted in Figure 1. The Precision values delivered by Winnow are more affected by the imbalance in the training collection than SVM, which is shown in Figure 4.

Depending on either optimizing Precision or Recall, different SVM parameter settings are determined in this study. Using a linear kernel with $C = 0.1$ and $J = 1.0$ achieves highest Precision, while optimal Recall is found with $C = 0.25$ and $J = 10.0$. Table 7 lists those settings in combination with the achieved Precision, Recall and F1 values. On a corpus of 500k patent documents, Precision tuning exceeds the baseline experiments by 6.34%, Recall tuning even by 16.32%.

Due to its robustness, SVM learning can be tuned on a small corpus and does not take up too much time if a careful grid search is applied. Keeping the cost parameter (J) low improves Precision, whereas a bigger J maximizes Recall. This finding can be extended to tuning other types of text corpora. A low cost parameter means allowing fewer mistakes (Precision) but therefore reduces the number of positive examples being retrieved (Recall). The complexity parameter (C) can be chosen in the interval $[0, 2]$.

Although detailed results on Balanced Winnow tuning could not be included due to the space limitations, it was also a part of the study. Low α-values of 1.01 and 1.02 along with 5 to 8 learning iterations yield optimal Precision/Recall on *itabsdesc* 500k, while α-values of 1.1 and 1.05 yield optimal Precision/Recall on *itabsdesc* 100k. Comparing with SVM two important differences can be observed. Firstly, Winnow tuning is not robust on different training collections. Secondly, the gap between Precision and Recall issignificantly smaller using the same parameter setting.

[8]Optimal Precision/Recall settings are given in Table 6. Due to space limitations, more detailed tuning results could not be included.



(a) Precision (b) Recall

Figure 6: Grid Search of SVM parameters C and J. More reddish (darker) colors denote low Precision/Recall, while brighter colors indicate high Precision/Recall.

To summarize the most interesting conclusions, all LCS Winnow experiments reveal moderate to low micro-averaged Precision and Recall results as well as moderate robustness in case of an increasing corpus size. *itabsdesc* significantly outperforms all other document representations. Parameter tuning showed that same parameter setting improves both Precision (72.23 %) and Recall (78 %).

The SVM experiments, on the other hand, achieve high Precision over all sample collections with an exceptional balancing of the Precision over all sub classes. SVM is highly robust in terms of number of sub classes to be learned. *itabsdesc* significantly outperforms all other document representations, while parameter tuning is indispensable.

The next step in this study is the development of automatic tuning methods. Depending on the requirements of the user, specific Precision- or Recall-directed tuning can be realized. Another issue that deserves attention in future investigation is the use of semantic information in the creation of the index. The incorporation of such linguistic information is the next component that will be added to the LCS.

7. REFERENCES

[1] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[2] C. J. Fall, A. Törcsvári, K. Benzineb, and G. Karetka. Automated categorization in the international patent classification. *SIGIR Forum*, 37(1):10–25, 2003.

[3] A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.

[4] M. A. Hearst. Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, 1998.

[5] M. Iwayama, A. Fujii, and N. Kando. Overview of classification subtask at NTCIR-5 patent retrieval task. In *Proceedings of the Fifth NTCIR Workshop*, Tokyo, Japan, 2005.

[6] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142, London, UK, 1998. Springer-Verlag.

[7] T. Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[8] C. H. A. Koster, M. Seutter, and J. Beney. Multi-classification of patent applications with Winnow. In *In Proceedings PSI 2003*, volume 2890, pages 545–554, LNCS, 2003. Springer.

[9] L. S. Larkey. A patent search and classification system. In *Proceedings of DL-99, 4th ACM Conference on Digital Libraries*, pages 179–187. ACM Press, 1999.

[10] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, April 1988.

[11] H. Nanba, A. Fujii, M. Iwayama, and T. Hashimoto. Overview of the patent mining task at the NTCIR-7 workshop. In *Proceedings of the Seventh NTCIR Workshop Meeting*, Tokyo, Japan, 2008.

[12] C. Peters and C. H. A. Koster. Uncertainty-based noise reduction and term selection in text categorization. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research*, pages 248–267, London, UK, 2002. Springer-Verlag.

[13] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[14] V. N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[15] Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, New York, NY, USA, 1999. ACM.

[16] F. Zaccá and M. Krier. Automatic categorisation applications at the European Patent Office. *World Patent Information*, 24:187–196, 2002.

Sentence Compression for Dutch Using Integer Linear Programming

Jan De Belder
Department of Computer Science
K.U.Leuven, Belgium
jan.debelder@cs.kuleuven.be

Marie-Francine Moens
Department of Computer Science
K.U.Leuven, Belgium
sien.moens@cs.kuleuven.be

ABSTRACT

Sentence compression is a valuable task in the framework of text summarization. In this paper we compress sentences from news articles taken from Dutch and Flemish newspapers using an integer linear programming approach. We rely on the Alpino parser available for Dutch and on the Latent Words Language Model. We demonstrate that the integer linear programming approach yields good results for compressing Dutch sentences, despite the large freedom in word order.

1. INTRODUCTION

Since the end of the 20th century, the compression of texts has been an active research topic in natural language processing (see for example the Document Understanding Conferences [17], and the more recent Text Analysis Conferences [21]). As this is a very difficult problem, it has often been reduced to the summarization of individual sentences, commonly referred to as sentence reduction [12] or sentence compression. This summarization task is the easiest in a word deletion setting, where we remove words from the original sentence, while maintaining a grammatical and coherent sentence that conveys the most important information [13]. For the Dutch language, the research in this area is limited. There has been some work on the summarization of documents in [1]. The compression of individual sentences has only been approached from a subtitle generation viewpoint [23] [24] [5] and a headline generation viewpoint [6]. In this paper, we investigate a generic method for sentence reduction, based on integer linear programming [4]. Required for this method are a language model, a parser, and a integer linear programming (ILP) solver.

The ILP approach operates by viewing sentence compression explicitly as an optimization problem. With a binary decision variable for each word in the original sentence, indicating whether or not it should be in the compressed sentence, the ILP solver finds an assignment for these variables that maximizes the probability of the sentence in the language model. In order to create well-formed summary sentences, the compression model might incorporate additional constraints that use grammatical rules of the language. As the most interesting information is most likely not very prominent in the language model, there is also need for a way of incorporating this information in the compressions. This is the function of the significance model.

In the next section we give an overview of relevant background work. Section 3 shortly introduces the tools we used for Dutch. Section 4 describes the main ideas of the integer linear programming approach. Our experimental setup can be found in section 5, and section 6 reports on the results. Finally, we give our conclusions and indications for future work in section 7.

2. BACKGROUND

Summarization or compression of text is a useful, but non-trivial application of natural language processing. Currently, there are several settings being researched that include the summarization of single documents [20], the summarization of multiple documents [25], and the summarization of single sentences. In this paper, we address the last setting.

Nearly all approaches of sentence compression rely on word deletion in such a way that the result is still a grammatical sentence, and conveys the most important information of the original sentence. A common application is headline generation based on the content of a larger text. By looking for headlines that are a subsequence of words in the first sentence of a news article, a sentence compression corpus can automatically be constructed. The offset for this approach was given in [13]. These authors used a parallel corpus of compressed and original sentences based on the Ziff-Davis corpus of news articles in the computer technology domain. The authors evaluated two compression methods. A noisy channel model considers an original sentence as the compressed sentence to which noise has been added. It assigns the most likely compression to the full sentence using Bayes rule, where the probability of a noisy component given a summary sentence is learned from the training data. The decision based model learns the discriminative reductions of the parse tree with a decision-tree learner based on the training data. The noisy-channel model is, however, not directly applicable for Dutch, due to lack of a Probabilistic Context Free Grammar. The decision based model has the disadvantage that the desired amount of compression cannot be given as a parameter.

In [6], headline generation was studied for the Dutch language. The method takes inspiration from the linguistically motivated Hegde trimmer algorithm [10], which employs rules to reduce the parse tree of a sentence, but learns the rules automatically using Transformation Based Learning, an error-driven approach for learning an ordered set of rules. The corpus that was used originates from Dutch news articles with matched headlines, taken from the Twente News Corpus.

Another setting in the compression of single sentences is the generation of subtitles for broadcasts. This is the case that has been mostly studied for Dutch [23] [24] [5]. These methods are based on shallow parsing and most of them require a parallel corpus for training. However, recent work [15] has shown that a word deletion approach is not very suited for subtitle generation.

There are also a few unsupervised approaches for sentence compression. [11] summarize the transcription of a spoken sentence, given a fixed compression rate. They use dynamic programming to find an optimal scoring solution, that takes a language model and the confidence of the speech recognizer into account. [22] define a semi-supervised and unsupervised version of the noisy channel model of [13]. [4] use an integer linear programming approach, which is applicable for any language, given the availability of a parser. This is the method that we will discuss, use, and modify in the remainder of this paper.

3. LANGUAGE TOOLS

In this section we describe the tools we used for constructing our Dutch sentence compression system.

3.1 Parsing

For parsing the Dutch sentences, we use the Alpino parser [2]. The Alpino system is a linguistically motivated, wide-coverage grammar and parser for Dutch in the tradition of Head-Driven Phrase Structure Grammars. It consists of about 800 grammar rules and a large lexicon of over 300,000 lexemes and various rules to recognize special constructs such as named entities, temporal expressions, etc. The aim of Alpino is to provide computational analysis of Dutch with coverage and accuracy comparable to state-of-the-art parsers for English. It is freely available for download.[1]

3.2 Latent Words Language Model

The Latent Words Language Model (LWLM) models the contextual meaning of words in natural language as latent variables in a Bayesian network [8]. In a training phase the model learns for every word a probabilistic set of synonyms and related words (i.e. the latent words) from a large, unlabeled training corpus. During the inference phase the model is applied to a previously unseen text and estimates for every word the synonyms for this word that are relevant in this particular context. The latent words help to solve the sparsity problem encountered with traditional n-gram models, leading to a higher quality language model, in terms of perplexity reduction on previously unseen texts [9]. In this article the model is trained on a 25m token corpus, consisting of Dutch newspaper articles.

[1] http://www.let.rug.nl/vannoord/alp/Alpino/

4. INTEGER LINEAR PROGRAMMING APPROACH TO SENTENCE COMPRESSION

In this section we will lay out the sentence compression method based on integer linear programming, following the line of work in [4]. We will start by shortly explaining what integer programming is, and how the basic method works by maximizing a language model probability. There are also extra constraints needed to make sure that a meaningful and grammatical sentence is obtained. In section 4.4 we discuss the significance model, that ensures that the generated compressions also contain topics of interest.

4.1 Integer Linear Programming

Integer linear programming is a restricted case of linear programming, where the values of the variables are limited to be only integers, instead of any real number. Linear programming tries to maximize (or minimize) an objective function, by searching for optimal values for the variables that constitute the objective function. This objective function is a linear combination of these variables, hence the name. The finding of an optimal combination of values is usually constrained. These constraints ensure that the variables cannot be infinitely large, and that the value of one variable can influence the other variables.

Integer programming has been used often in Natural Language Processing, for many different tasks. In many situations, NLP constitutes searching in very large hypothesis spaces, like packed forests of parse trees [16]. Other applications include a.o. coreference resolution [7] and semantic role labeling [19]. Integer linear programming, a technique that has often been used in optimalisation theory for many decades, is very well suited for these kind of problems, as it enables us to efficiently search for the optimal solution, and at the same time incorporate constraints on a global scale.

4.2 Integer Programming For Sentence Compression

Given a sentence $W = w_1 \dots w_n$, our goal is to obtain a sentence W^*, with a reduced number of words. For a sentence $W = w_1 \dots w_n$, we first need decision variables to indicate whether or not w_i should be in the compressed sentence. We notate these variables with y_i, with a value of 1 if word w_i is in the compressed sentence, and 0 if it is not. For clarity, suppose we want the ILP solver to find a sentence that maximizes a unigram model, than the objective function would look like this:

$$max\ z = \sum_{i=1}^{n} y_i P(w_i),$$

with $P(w_i)$ being the unigram probabilities. This overly simple model is not adequate; a trigram model would have much better performance. This comes down to adding three additional types of variables. In short, we need n extra variables to indicate whether or not a word starts the sentence (p_i), and $\frac{n \cdot (n-1)}{2}$ decision variables that indicate whether two words end the sentence (q_{ij}). Finally, there are $\frac{n \cdot (n-1) \cdot (n-2)}{6}$ variables needed to indicate whether a specific trigram $w_i w_j w_k$

is in the sentence (x_{ijk}). These three types of variables are needed for constraints on the language model. For example, only one word can start the sentence, which translates to a constraint in the ILP model. Without these constraints, the ILP would set all variables to 1, and say that all words start the sentence. The complete list of constraints can be found in [4], but will not be repeated due to spatial constraints, and the fact that they are not required to understand to operations behind the method. The objective function of the integer linear programming problem is given in the following equation[2]:

$$\begin{aligned} max\ z &= \sum_{i=1}^{n} p_i P(w_i|\text{start}) \\ &+ \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} x_{ijk} P(w_k|w_i w_j) \\ &+ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} P(\text{end}|w_i w_j) \end{aligned} \quad (1)$$

4.3 Linguistic Constraints

The ILP model given above is language independent. However, the fact that a sentence has a high probability in a language model, does not make it a grammatical and fluent sentence. That is why there is the need to incorporate language specific grammatical information in the method.

The constraints described below are motivated from a linguistic and intuitive point of view and are often dependent on the language used. These constraints are based on a parse tree and the grammatical relations of a sentence, and can be used in combination with any parser. In [4], the Robust Accurate Statistical Parsing toolkit was used [3]. As described in section 3.1, for Dutch we are limited to the use of Alpino.

Modifier Constraints. It is often the case that determiners can be left out of the compression (especially in the case of headline generation). This still yields a grammatical sentence. The other way around, i.e. keeping the determiner but removing its head word, is not acceptable. This leads to the following constraint:

$$\begin{aligned} y_i - y_j &\geq 0 \\ \forall i,j : w_j &\in w_i\text{'s determiners} \end{aligned} \quad (2)$$

If a determiner w_j is in the compression, which corresponds to y_j having the value 1, the constraints force y_i to take the value 1 as well, causing the head word w_i to be in the compression.

Some determiners cannot be left out, especially when they change the meaning of their head word, and thus probably the meaning of the entire sentence. The most trivial one is the word 'not'. We also included the word 'none'. An important modifier for Dutch is the word *er*, which translates roughly as 'there'[3]. This constraint can be removed, but it generates more fluent sentences, rather than headline-style sentences. Possessive modifiers are also added to the list.

$$\begin{aligned} y_i - y_j &= 0 \\ \forall i,j : w_j &\in w_i\text{'s determiners} \wedge \\ w_j &\in \text{(not, none, possessives, '}er\text{')} \end{aligned} \quad (3)$$

Note that the difference between constraints 2 and 3 is in the sign of the equation: constraint 2 uses a $\geq$ sign to indicate that w_i can be in the compression by itself, but w_j can not. Constraint 3 uses an $=$ sign, which mean that either both w_i and w_j have to be in the compression or either none of them can be in the compression.

Argument Structure Constraints. The next constraints are needed for the overall sentence structure. Constraint 4 makes sure that if there is a verb in the compressed sentence, then so must be its arguments. The reverse also has to be true: if there is a subject from the original sentence taken for the compressed sentence, so must be the corresponding verb.

$$\begin{aligned} y_i - y_j &= 0 \\ \forall i,j : w_j &\in \text{subject/object of verb } w_i \end{aligned} \quad (4)$$

$$\sum_{i:w_i \in verbs} y_i \geq 1 \quad (5)$$

Constraint 5 requires that, if there is a verb in the original sentence, there should also be at least one in the compressed sentence.

One of the peculiarities of Dutch[4] are separable verbs that fall apart into their original parts, when you conjugate them. For example, *toepassen* (to apply), becomes in the first person singular *ik pas toe* (I apply). If a compressed sentence contains the stem of the separable verb, it should also include the separated part, and vice versa. The parser detects these separable verbs, so we can define the following constraint:

$$\begin{aligned} y_i - y_j &= 0 \\ \forall i,j : w_j &= \text{separated part of separable verb } w_i \end{aligned} \quad (6)$$

Furthermore we also require the predicative adjectives to be included together with their head, and the same for reflexive objects such as 'themselves'.

There are two other constraints needed for prepositional phrases and subordinate clauses in order to ensure that the introducing term is included, if any word from the phrase or clause are included (defined in equation 7). Subordinate clauses are those that begin with a wh-word, or with subordinating conjunctions such as 'after' or 'because'. The

[2]From here on we assume all probabilities are log-transformed

[3]For example in the sentence '*Er is melk in de koelkast*', which translates to 'There is milk in the fridge'. Sometimes this is not as clear. The sentence 'Something has to be done' translates to '*Er moet (has) iets (something) gedaan (done) worden (to be)*'.

[4]This is also common in German and Hungarian.

reverse should also hold (see equation 8).

$$\begin{aligned} y_i - y_j &\geq 0 \quad (7) \\ \forall i,j : w_j &\in \text{PP/SUB} \wedge \\ w_i & \quad \text{starts PP/SUB} \end{aligned}$$

$$\sum_{i:w_i \in \text{PP/SUB}} y_i - y_j \geq 0 \quad (8)$$

$\forall j : w_j$ starts PP/SUB

General Constraints. Alpino is able to detect multi word units (MWUs). These can be names of persons, such as *Minister Van Der Donck*, but also parts of expressions, such as *op wacht staan* (to stand guard). For simplicity we define a constraint that either all words of the MWU should be included, or none of them.

Related to the compression length, it is possible to define an upper and lower bound on the generated compression. Enforcing a length of at least l tokens is done with the following constraint:

$$\sum_{i=1}^{n} y_i \geq l \quad (9)$$

Defining an upper bound can easily be done by replacing the $\geq$ sign with $\leq$.

4.4 Significance Model

A probable side effect of relying on a language model to generate compressions, is that the model will prefer known words. This has as a consequence that the most important words in the sentence, for example names of persons, will not be likely to appear in the compression. The solution for this problem lies in a significance model. This model assigns a weight to every topic word in the sentence, with a topic word being a noun or a verb. The weights are based on several statistics, and calculated with the following equation:

$$I(w_i) = \frac{l}{N} f_i log \frac{F_a}{F_i} \quad (10)$$

where f_i and F_i are the frequencies of word w_i in the document and a large corpus respectively, F_a the sum of all topic words in the corpus. l is based on the level of embedding of w_i: it is the number of clause constituents above w_i, with N being the deepest level in the sentence[5]. To incorporate these weights in the objective function given by equation 1, the sum of equation 10 over the topic words can be simply added, resulting in the following equation:

$$\begin{aligned} max\ z &= \lambda \sum_{i=1}^{n} y_i I(w_i) + \sum_{i=1}^{n} p_i P(w_i|\text{start}) \\ &+ \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} x_{ijk} P(w_k|w_i w_j) \\ &+ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} P(\text{end}|w_i w_j) \quad (11) \end{aligned}$$

[5]Deeply embedded clauses tend to carry more semantic content. For example in the sentence 'U.S. officials said there has been a bombing', the embedded fragment 'there has been a bombing' contains the most important information.

The parameter λ weighs the importance of the language model versus the significance model, and can be estimated on a small set of training data.

5. EVALUATION

5.1 Data

The data consists of news articles written in Dutch, coming from major Belgian and Dutch newspapers and crawled from the Web pages of the news providers. We selected the websites and articles at random, to have a diverse set of texts. The articles date back to the beginning of 2008. We used a set of articles from 31/1/2008 and 1/2/2008 for development and training, and articles from 6/2/2008 and 8/2/2008 for the evaluation.[6] We manually segmented the articles into sentences, to ensure a clean dataset. The training and development data consisted of 40 articles, the evaluation data of 30.

Since the evaluation is done manually, as will be described in section 5.3, the amount of sentences that we can evaluate is limited. Here we took the first sentence of each article in the evaluation set, and limited these further to sentences that contain at least 15 tokens. This resulted in a set of 21 sentences, with an average length of 20.8 tokens, ranging over a diverse set of topics.

We used a different data set to train the Latent Words Language model. We took a 25 million token subset of the Twente News Corpus [18], from four different newspapers in the year 2005. The dictionary size was limited to 65.000 words. We also used this data to estimate the corpus frequency of the topic words, as described in equation 10. If a topic word was not present in the corpus, we estimated its weight as the average of the other topic words in the sentence.

5.2 Systems

For the evaluation we tested the system in four different settings, all based on the integer linear programming approach. The first system relies solely on the language model, and does not use any grammatical information. The second system does use the grammatical constraints. The third and fourth system both add the significance model, but with different values for the parameter λ. As described in section 4.4, this parameter weighs the importance of the significance model against the language model. During initial testing it became clear that it is very difficult to estimate this parameter. Values that work for some sentences yield lesser results on other sentences. It also has a significant influence on the length of the compression, where higher values for λ tend to generate longer sentences. Higher values cause the system to only include the topic words, while still being limited by the constraints, which results in using all the topic words without everything that is dictated by the constraints. For these reasons, we did the evaluation with two different values for λ: 0.75 and 1.5, that both had good empirical results on the development data.

[6]This difference in time was needed to ensure that no articles in the evaluation data overlapped with those in the development data.

Finally, we constrained the systems to generate compressions of at least 40% of the original length, by using the constraint in equation 9.

5.3 Evaluation

As is good practice in the testing of summarization systems, we opted for manual evaluation. We did two different experiments. In the first experiment, we presented the participants with a list of generated compressions, each from a different original sentence. We asked the participants to give a score for the grammaticality of each sentence, on a five point scale. In the second experiment the participants were given the original sentences together with the corresponding compressions, and they were asked to rate the compressions based on the retention of the most important information, again on a five point scale. The sets of sentences were generated at random: each set contained compressions from the different systems. Together with the four systems defined above, we added a manually constructed set of compressions made by one of the authors. The participants were told that all the sentences were machine generated. This allows us to compare the machine generated compressions with one made by a human, and define an upper bound on the performance that is achievable in a word-deletion setting. In total we had 15 participants, each grading 21 sentences based on grammaticality, and another 21 sentences on content.

Using the manually constructed set of compressions, we also calculated the ROUGE scores [14], as often applied in the DUC competitions. We used the ROUGE-2, ROUGE-L, and ROUGE-SU4 metrics, that assign scores based on bigram co-occurrences, the longest common subsequence, and skip-bigrams in combination with unigrams respectively.

6. RESULTS

6.1 Human Evaluation

6.1.1 Grammaticality

The results on the manual evaluation can be found in table 1. From the column that reports on the grammaticality of compressions, it is clear that the grammatical constraints are necessary. The system that uses only the language model to generate compressions, did not came up with many meaningful sentences. This is very likely due to the limited size of the language model used. The systems that do use the grammatical constraints usually come up with a grammatical compression. The median of grammaticality scores is 4, for each of the three systems that used the grammatical constraints. Annotators often punished the compressions due to not incorporating the determiners, which generates more headline-like compressions. The leaving out of commas was also a cause for lower ratings. In one case none of the systems was able to include the main verb and subject, which did not happen when using a longer minimum compression length. The biggest problem is the needed inversion of a verb and a subject when a prepositional phrase is removed from the beginning of the sentence. Switching the verb and the subject in a sentence would require substantial modifications to the ILP method. The grammatical information from the parse tree would not just lead to the adding of more constraints, but to the addition of more decision variables and a modification of the objective function, which we leave for further research.

6.1.2 Significance Model

Looking further we can see that the significance model has an impact on the information retention, although this is rather limited. Despite the fact that the last system ($\lambda = 1.5$) generates on average sentences that are 14% longer, this has little influence on the scores given by the participants of the experiment. The reason for this is that the most important information usually takes the role of subject or object, and is thus already required to be in the compression. The difference in score between the best system and the human made compressions is larger than for the grammaticality, but it should be noted that the human made compressions are on average almost 10% longer.

6.2 Automatic Evaluation

From the results in table 2 we can conclude that the automatic evaluation measures all follow the human judgment. The version of the system with the significance model ($\lambda = 1.5$) scores the best, which indicates that this model generates compressed sentences that are the closest to the handcrafted summaries.

6.3 Discussion

In general, the method performs rather well. When compared to the human made summaries, the best model only scores ± 1 point lower, both on grammaticality and content. We also tested whether the human made summaries were possible to create by the ILP method, using the grammatical constraints imposed. In 12 out of the 21 cases, this was not possible. Often the cause was a small error in the parsing process, especially in the case of PP-attachments.

Another related problem can be found in the compression of names. Often these are accompanied by a description of their function, for example 'The French president Sarkozy'. Without loss of information, this can easily be reduced to 'Sarkozy'. But when talking about the Serbian president Boris Tadić, the participants of the experiments preferred the descriptive compression 'the Serbian president' over the actual name 'Boris Tadić'. This problem is not only present in Dutch, but in summarization in general.

In these experiments we defined specific values for λ and a specific lower bound on the sentence length, in order to obtain just one compression from every system. However, the systems can easily generate an entire set of compressions by varying the parameters, more often than not generating better compressions than given here. As the solving of the ILP problem is several orders of magnitude faster than parsing the sentence with the Alpino parser, it is our opinion that the determination of the best compression, given a set of possible compressions, can better be handled in a later stage.

7. CONCLUSION

In this paper we have presented a sentence compression method for Dutch, a free word order language. We used an integer linear programming approach that finds a compression by maximizing the language model probability, while constrained to be grammatical. For this we used Alpino, a

[7]We define the average compressed rate as the average percentage of words retained in the compression.

System	Avg. Comp. Rate[7]	Grammar	Information
Human	66.9%	4.71 ± 0.40	4.43 ± 0.53
LWLM	43.0%	1.29 ± 0.54	1.26 ± 0.30
LWLM+Gram	43.3%	3.45 ± 1.47	3.14 ± 1.31
LWLM+Gram+Sig ($\lambda = .75$)	49.0%	3.81 ± 1.38	3.19 ± 1.67
LWLM+Gram+Sig ($\lambda = 1.5$)	57.5%	3.98 ± 1.12	3.41 ± 1.19

Table 1: Manual evaluation results of the four systems and the handcrafted summaries, on grammaticality and information retention of the compressions.

System	ROUGE-2	ROUGE-L	ROUGE-SU4
LWLM	0.240	0.569	0.341
LWLM+Gram	0.431	0.650	0.469
LWLM+Gram+Sig ($\lambda = .75$)	0.472	0.697	0.505
LWLM+Gram+Sig ($\lambda = 1.5$)	0.508	0.712	0.530

Table 2: Automatic evaluation results with the ROUGE toolkit, using the handcrafted summaries as a gold standard.

parser for Dutch, and the Latent Words Language Model. We needed extra language-specific constraints on the generated compressions to maintain the meaning, which we accomplished by using the output of the parser. We also identified some shortcomings, by checking whether the handcrafted compressions can be generated under the grammatical constraints, which was not always the case.

The next step is to extend the integer linear programming approach to allow for words to swap places, allowing the model to generate more grammatical compressions. We also believe that the meaningful compression of person names with their description could be learned from training data, in addition to this otherwise unsupervised method.

8. ACKNOWLEDGMENTS

This research is funded by the Dutch-Flemish NTU/STEVIN project *DAISY*[8] (ST 07 015) and the EU project *PuppyIR*[9] (EU FP7 231507).

9. REFERENCES

[1] R. Angheluta, R. De Busser, and M.-F. Moens. The use of topic segmentation for automatic summarization. In *Proceedings of the ACL-2002 Workshop on Automatic Summarization*. Citeseer, 2002.

[2] G. Bouma, G. Van Noord, and R. Malouf. Alpino: Wide-coverage computational analysis of Dutch. In *Computational Linguistics in the Netherlands 2000. Selected Papers from the 11th CLIN Meeting*, 2001.

[3] T. Briscoe, J. Carroll, and R. Watson. The second release of the RASP system. In *Proceedings of the COLING/ACL*, volume 6, 2006.

[4] J. Clarke and M. Lapata. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31(1):399–429, 2008.

[5] W. Daelemans, A. Hothker, and E. Sang. Automatic sentence simplification for subtitling in Dutch and English. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1045–1048. Citeseer, 2004.

[6] D. de Kok. Headline generation for Dutch newspaper articles through transformation-based learning. Master's thesis.

[7] P. Denis and J. Baldridge. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of NAACL HLT*, pages 236–243, 2007.

[8] K. Deschacht and M.-F. Moens. Semi-supervised semantic role labeling using the latent words language model. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*.

[9] K. Deschacht and M.-F. Moens. The Latent Words Language Model. In *Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning*, 2009.

[10] B. Dorr, D. Zajic, and R. Schwartz. Hedge Trimmer: a parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pages 1–8. Association for Computational Linguistics Morristown, NJ, USA, 2003.

[11] C. Hori and S. Furui. Speech summarization: an approach through word extraction and a method for evaluation. *IEICE Transactions on Information and Systems*, 87:15–25, 2004.

[12] H. Jing. Sentence reduction for automatic text summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 310–315, 2000.

[13] K. Knight and D. Marcu. Statistics-based summarization-step one: Sentence compression. In *Proceedings of the National Conference on Artificial Intelligence*, pages 703–710. MIT Press, 2000.

[14] C. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26, 2004.

[15] E. Marsi, E. Krahmer, I. Hendrickx, and W. Daelemans. Is sentence compression an NLG task? In *Proceedings of the 12th European Workshop on*

[8] http://www.cs.kuleuven.be/ liir/projects/daisy/

[9] http://www.puppyir.eu

Natural Language Generation, pages 25–32. Association for Computational Linguistics, 2009.

[16] A. Martins, N. Smith, and E. Xing. Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACLŠ09), Singapore*, 2009.

[17] A. Nenkova. Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1436. MIT Press, 2005.

[18] R. Ordelman, F. de Jong, A. van Hessen, and H. Hondorp. Twnc: a multifaceted Dutch news corpus. *ELRA Newsletter*, 12(3/4):4–7, 2007.

[19] D. Roth and W. Yih. Integer linear programming inference for conditional random fields. In *Proceedings of the 22nd international conference on Machine learning*, page 743. ACM, 2005.

[20] Z. Teng, Y. Liu, F. Ren, and S. Tsuchiya. Single document summarization based on local topic identification and word frequency. In *Artificial Intelligence, 2008. MICAI'08. Seventh Mexican International Conference on*, pages 37–41, 2008.

[21] Text Analysis Conference (TAC). http://www.nist.gov/tac/.

[22] J. Turner and E. Charniak. Supervised and unsupervised learning for sentence compression. *Ann Arbor*, 100, 2005.

[23] V. Vandeghinste and Y. Pan. Sentence compression for automated subtitling: A hybrid approach. In *Proceedings of the ACL Workshop on Text Summarization*, pages 89–95, 2004.

[24] V. Vandeghinste and E. Sang. Using a parallel transcript/subtitle corpus for sentence compression. In *Proceedings of LREC 2004*. Citeseer, 2004.

[25] D. Wang, S. Zhu, T. Li, and Y. Gong. Multi-document summarization using sentence-based topic models. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 297–300, Suntec, Singapore, August 2009. Association for Computational Linguistics.

Re-ranking based on Syntactic Dependencies in Prior-Art Retrieval

Eva D'hondt, Suzan Verberne, Nelleke Oostdijk, Lou Boves
Centre for Language Studies & Information Foraging Lab,
Radboud University Nijmegen,
(e.dhondt|s.verberne|n.oostdijk|l.boves)@let.ru.nl

ABSTRACT

In this paper we present an experiment using syntax (in the form of dependency triplets) to rerank retrieval results in the patent domain. This work is a follow-up experiment of our participation in the first CLEF-IP track, which focussed on prior art retrieval. We shall first describe the work done in our participation to the CLEF-IP track and then go on to show why improving Mean Average Precision (MAP) is important to the patent searchers community. We then introduce an additional reranking step to our BOW retrieval approach which is based on syntactic information. Using syntactic structures called Dependency Triplets as index terms we perform a second retrieval step within the retrieved result sets and examine if the ranking of the relevant documents (captured by the MAP score) can be improved for prior art search.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search

General Terms

Dependency Triples

Keywords

Prior Art, patent retrieval, syntactic units

1. INTRODUCTION

Patent retrieval is a rising research topic in the western Information Retrieval (IR) community. Though it already was the topic of workshops in SIGIR 2000 and ACL 2003 and has been a recurring track in the NTCIR workshops since 2002, it has not gathered a lot of attention from the western Information Retrieval community, mainly because the document collections of the NTCIR workshops are more focussed on Asian languages. In 2009, however, the first Patent Retrieval track with a focus on European languages (CLEF-IP)[1] was organized by the Information Retrieval Facility (IFR) as part of the CLEF 2009 evaluation campaign.[2] The general aim of the track is to explore patent searching as an information retrieval task and bridge the gap between the IR community and the world of professional patent search.

The goal of the 2009 CLEF-IP track was 'to find patent documents[3] that constitute prior art[4] to a given patent' [20]. In this retrieval task each topic query was a (partial) patent document which could be used as one long query or from which smaller queries could be generated. The track featured two kinds of tasks: In the Main Task prior art had to be found in any one (or combination) of the three following languages: English, French and German; three optional subtasks used parallel monolingual topics in one of the three languages. In total 15 European teams participated in the track. Because of this high participation rate, the CLEF-IP track will be sure to continue next year.

At the Radboud University of Nijmegen we decided to participate in the CLEF-IP track because it is related to the focus of the Text Mining for Intellectual Property (TM4IP) project[15] that we are currently carrying out. In this project we investigate how linguistic knowledge can be used effectively to improve the retrieval process and facilitate interactive search for patent retrieval. Because the task of prior-art retrieval was new to us, we chose to implement a baseline approach to investigate how well traditional IR techniques work for this type of data and where improvements would be most effective. These results will effectively serve as a baseline for further experiments as we explore the influence of using dependency triplets[5] for various IR tasks on the same patent corpus.

[1] http://www.ir-facility.org/research/evaluation/clef-ip-09/overview

[2] See http://www.clef-campaign.org

[3] In this paper we use the following terminology: a 'patent document' is physical document which is a version of a patent (application) at a certain point in time; A 'patent' is a set of documents that carry the same patentID code. This is explained in more detail in section 3.1.

[4] Prior art for a patent (application) means any document (mostly legal or scientific) that was published before the filing date of the patent and which describes the same or a similar invention.

[5] A dependency triplet is a unit that consists of two open category words and a meaningful grammatical relation that binds them.

In the CLEF-IP task we used a standard retrieval approach based on keyword matching, using the Lemur retrieval engine and the TF-IDF ranking algorithm. This baseline run achieved moderate results compared to the other participants (Recall@100= 0.22 and MAP=0.054). Overall, the results of all participants were rather low, compared to retrieval results in other tasks: Recall@100 ranged from 0.58 to 0.02 and Mean Average Precision (MAP)[6] from 0.11 to 0.00 (with one outlier: the run submitted by the Humboldt University which achieved 0.27). These general results will be further discussed in section 2.4.

The MAP score and the Recall score are the two most important measures for patent retrieval [3]. Recall must be very high, because for patent searchers it is extremely important to find ALL relevant documents. The financial repercussions of an incomplete prior art search can be severe, even if the patent has already been granted. But while recall is important, it is also clear that patent searchers cannot afford to process large result sets comprising thousands of patents that have to be browsed through completely: Patent retrieval is a highly interactive search task where the information need is constantly modified throughout the search. Finding a particularly relevant document at an early stage of the search will enhance the effectiveness of the remainder of the search task. (For example, by adding new keywords, IPC[7] codes, etc. gained from this document to the query.) Therefore, improving the ranking of the relevant documents in the result set is important to the patent searcher.

There is evidence in the IR literature (see section 2.1) that using dependency relations to rerank a small, already retrieved set of documents can be very successful for ad-hoc document retrieval and QA. The dependency model used in the TM4IP project differs from most other dependency models in that it is developed for IR purposes and is therefore linguistically less detailed than other models. In the project we are currently developing the AEGIR parser, a rule-based dependency parser which is geared towards the specifics of the language used in patents and which is more robust than other general-language parsers. This parser generates dependency triplets from the input text, which are then -in turn- used as index terms in the interactive retrieval system (also under development).

In this paper we focus on improving MAP of the result list, produced in the CLEF-IP experiment, by adding an extra step. To this end we perform an additional reranking operation on the result set using syntactic information in the form of dependency triplets[8]

2. BACKGROUND

2.1 Syntax-based retrieval

In Information Retrieval the bag-of-words approach (BOW) is the approach most frequently used for all types of IR tasks. It is attractive to researchers because it makes the model simple, easily manageable and comprehensible. However, a recurring criticism on the BOW approach is the fact that by splitting the text up into single terms, the model does not take into account the immediate context of the terms and subsequent relations between terms. For example, a simple BOW-based retrieval system cannot differentiate between the following two queries: *bank terminology* and *terminology bank* [29]

In the last two decades, several approaches have been developed that use larger retrieval units, namely phrases. Phrases can be defined by their statistical properties or syntactic characteristics or a combination of two. The most successful statistical approaches are proximity-based phrase indexing [10], the n-gram retrieval model[9] [24] and the term dependency modelling approach [11], [18]. These approaches focus on taking context into account and are able to capture some (dependency) relations between terms on the basis of their collocation frequencies. However, they typically fail on long distance dependencies.

The more linguistically-motivated approaches, such as [23], [26], [2] have focussed on extracting syntactic units from the text using linguistic information. These phrases can either take the form of a head-modifier pair or a (partial) dependency tree. Several studies have investigated the effect of using syntactic versus statistical phrases as index terms: [10],[16], [13], [1] found that there is only a small improvement when syntactic relations are taken into account in the retrieval process. Syntactic phrases have been found to be useful, however, for improving the ranking of the results found by a BOW approach, at least for ad-hoc search [4] and QA [8],[28]. [6] reports that the longer the queries, the more useful NLP techniques like extracting dependency pairs can become, though he adds that (at least for ad hoc search) the benefit is limited.

[25] argues that one of the reasons for the disappointing results in dependency-based retrieval could be the fact that the earlier systems did not take the *variability of the structure* of the syntactic phrase into account: In a noun phrase like *World Bank criticism* a syntactic phrase that contains a compound like 'World Bank' is a much more important retrieval unit than *Bank criticism* and should be given more weight as an index term. [19] remarks that part of the discouraging effect of phrases in text retrieval stems from the fact that they must be normalized to a standard form (in order to rise above syntactic and lexical variation). Such transformations are complex and prone to errors. The removal of function words (e.g. prepositions, determiners, ..) plays an important role in this normalisation process [10].

2.2 Syntax in Patent Retrieval

The majority of the search engines used by the patent search community today are keyword-based, using a general-purpose text search engine. Academic research on patent retrieval has mainly been focussed on the relative weighing of the index terms [17] and on exploiting the patent document structure to boost retrieval [17]. There is a lot of attention for query reformulation at the moment, as could be seen in the

[6]MAP is a measure of how high the relevant documents appear in the result list, measured over all queries.

[7]The International Patent Classification, used by all major patent offices.

[8]In our approach to dependency triplets we consider them as single index units, not as relationships between two separate index terms.

[9]For an overview of related articles and patents, see http://www.cs.umbc.edu/ngram/

CLEF-IP track where 5 out of 14 teams actively explored different query term selection and query reformulation strategies. For an overview of the state of the art in academic and commercial systems, see [5].

There are not many approaches in the patent domain that use syntactic phrases or structures comparable to our approach which we will explain in section 2.3. Systems like [9] and [7] perform a combination of syntactic and semantic analysis on the documents and use the results to generate concept units. The only purely syntactic approach is [21], who uses deep linguistic analysis in the form of predicate-argument analysis (implying semantic role labelling) to improve readability of the claims section. Her system is the first step in a suggested patent summarization method.

2.3 The CLEF-IP track

In answer to a growing demand from the patent searcher community for reliable and improved patent search engines the first CLEF-IP track was organised by the IRF. As was explained in section 1, it aims to bring the IR community and the world of professional patent search closer together to create new and innovative retrieval systems. The first track can be considered a major success as it received a lot of interest from the IR community and –in turn– presented the IR community with a patent corpus of significant size within an integrated and single IR evaluation collection. The results of the participating groups in the patent track yielded some interesting insights into the particulars of patent retrieval: as mentioned above the overall precision and recall results in this task were quite low (average Precision@100= 0.02, average Recall@100=0.38, average MAP = 0.07, except for one outlier) compared to the results in other retrieval tracks.

There are a number of reasons for these low scores: First of all, some of the documents were 'unfindable': 17% of the patent documents in the collection contained so little information, e.g. only the title which is poorly informative for patent retrieval [27], that they could not be retrieved. Secondly, the relevance assessments were based on search reports and the citations in the original patent only. This means that they were conceptually-based and not text-based and may therefore have been too limited[10]. Finally, in order to perform retrieval on the patent level, instead of the document level, some of the participating groups created 'virtual patents': for each field in the patent the most recent information was selected from one of the documents with that patentID. These fields were glued together to form one whole 'virtual' patent. It is, however, not necessarily true that the most recent fields are the most informative [27]. This selection operation may have resulted in a loss of information. However, even without these impediments, it is clear that patent retrieval is a difficult task for standard retrieval methods.

2.4 The TM4IP project

At the Radboud University Nijmegen, we are currently involved in the Text Mining for Intellectual Property (TM4IP) project[15], which is directed at developing an approach to interactive patent search using syntactic structures in the form of dependency triplets as search terms and for computing the relevance ranking. While the idea of using (partial) syntactic phrases as index terms is not new (see section 2.1), the dependency model used in TM4IP differs from previous attempts in that it is based on the notion of *aboutness* to suit retrieval purposes. Aboutness is a difficult concept to define and has many different interpretations in the literature. In IR it is defined as follows: the user of a retrieval system expects the system, in response to a query, to supply a list of documents which are *about* that query. Practical retrieval systems using single words as terms are based on an extremely simpleminded notion of aboutness. For our system, the concept of *aboutness* implies that we do not allow any words in the dependency triplets that have no classificatory value as keywords (by themselves) [15].

In this project a rule-based dependency parser has been constructed that is now being tuned to deal with English technical texts. In the near future, this parser will also incorporate frequency information on words and on triplets and will thus become a hybrid parser. This parser generates dependency triplets (structured units, containing word forms and dependency relations) from the input text, which are then –in turn– used as index terms in the retrieval system. The aim of the project is to successfully use linguistic knowledge (in the form of dependency triplets) to improve the retrieval process and facilitate interactive search for patent retrieval. We have already achieved good results using the dependency triplets as basic units for the classifier for patent documents that is also a part of our system [14]. Using dependency triplets as classification terms, we reached a high accuracy in the (pre)classification of patent applications in their correct IPC classes.

The full dependency triplet-based patent search system is still under development. Therefore, in this paper we investigate the effect of using dependency triplets for improving the relevance ranking of documents that have been retrieved by some conventional search engine. Literature shows that re-ranking with dependency triplets can be successful (see section 2.1).

3. METHODOLOGY

3.1 Data

The CLEF-IP corpus consists of European Patent Office (EPO) documents that have been published between 1985 and 2000, covering English, French, and German patents. In total, the corpus contains 1,958,955 patent-documents pertaining to 1,022,388 patents (75GB) as one patent can consist of multiple XML files: A patent can consist of several documents that were produced at different stages of a patent realization.[11] For example, a so-called A2 document (the patent application in its barest form, submitted at the beginning of the patent application process) can contain only a title and perhaps an abstract, while a B1 document (a granted patent, usually finished three years after the initial application) will contain a title, abstract, claims and description section.

The heterogenity of the corpus has certain implications for

[10] http://www.clef-campaign.org/2009/working_notes/CLEF-2009WNContents.html

[11] For an overview of the patent kind codes used in the corpus, see http://www.delphion.com/help/kindcodes under EPO.

the search process: While it seems preferable to search only in the B1 documents, this would exclude a large number of documents from the search that could be relevant while searching for prior art: some patents only consist of an A2 document.

In the CLEF-IP 2009 track the participating teams were provided with 4 different sets of topics (S,M,L,XL). We opted to do runs on the smallest set (the S data set) for both the Main and the English task. This set contained 500 topics. Because the information in these topics was different for both tasks[12] we focussed on the data that was available in all the topics: the English claims sections. As only 70% of the CLEF-IP corpus contained English claims, this means that a substantial part of the corpus could not be retrieved.[13] By reducing the patent documents to the claims sections only, we gained consistency (all the documents to be retrieved have the same style of writing and are not empty). Even so, improving consistency in the way we did comes with a price. We might have thrown away that part of the document containing the relevant information. In the patent retrieval literature, however, there is evidence [12],[22] that the claims section is the more informative part of the patent document. Nonetheless, we may wonder if –for the reranking experiment– limiting our document set to claims text only will not have an adverse effect on the generation of the index terms (dependency triplets): It might be that this will put an additional strain on the parser, as the language in claims is notoriously difficult to read and highly complex, therefore quite difficult to parse correctly.

3.2 Baseline approach

3.2.1 Queries

After removing punctuation and stopwords we took all remaining words in the claims section together as one long query (weighted in retrieval with TF-IDF). No stemming was conducted.

3.2.2 Indexing and Retrieval using Lemur

We extracted the claims sections from all English patent documents in the corpus and removed all XML markup from the texts by means of a preprocessing script. Since there may be multiple documents that carry the same patent-ID, we concatenated the claims sections pertaining to one patent ID into one document in the index file. We saved all patent claims in the Lemur index format with the patent IDs as DOCIDs. They were then indexed using the BuildIndex function of Lemur with the indri IndexType and a stop word list for general English. The batch retrieval was then performed using TF-IDF.

3.3 Reranking experiment

3.3.1 Data selection

[12]Some of the topics for the Main Task contained the abstract content as well as the full information of the granted patent except for citation information, while the topic patents for the English Task only contained the title and claims elements of the granted patent [3].

[13]Of the 30% percent that could not be retrieved by our system, 7% were documents that only had claims in German or French but not in English, 6% only contained a title and abstract, usually in English and 17% only contained a title.

In the baseline experiment we retrieved 100 results for each of the 500 topics but because some documents were attributed to multiple topics we only retrieved a total of 39,802 unique documents. In total the retrieved documents contained around 52 million words. The average sentence length in these document was 49 words and the longest sentence in the retrieved documents consisted of 451 words.

In the reranking experiment we took all 100 documents of the result set (per topic), parsed them (see 3.3.2), used Lemur to create a separate index containing all the triplets of the retrieved documents per topic and performed a second retrieval on these indices. On average, the result sets contained around 85,000 words each.

We chose this set-up to compare the impact of dependency triplets in the ranking of the documents. For each topic, the exact same hundred documents are available in the index that were found (for that topic) in the baseline experiment. Consequently, the same (number of) relevant documents will be found in the second retrieval step. Therefore, recall and precision will remain the same in the second experiment and only the ranking of the (relevant) documents (measured in MAP) can be subject to change.

3.3.2 Pre-processing

We parsed the topics and the 39,802 retrieved documents of the CLEF-IP corpus using the AEGIR parser (version 1.1). The grammar from which the parser was generated comprises some 200 rules. The dependency model used by the parser has the following format: `[term`$_1$`, relator, term`$_2$`]`. The sentence 'The system consists of four separate modules' will be turned into the following triplets: `[system,SUBJ,consists]`, `[consists,PREPof,modules]`, `[modules,ATTR,separate]`, `[modules,QUANT,four]`. Our dependency model is based on the notion of aboutness: with a few exceptions only open category members are allowed as head or modifier. In the example given above, the determiner 'the' is not allowed into the triplets. We used a small set of relators which mirror basic semantic relations :

- SUBJ(ect):
  - 'The method describes' `[method,SUBJ,describes]`;
  - '(Object) claimed by Microsoft' `[Microsoft,SUBJ,claimed]`;
- OBJ(ect) :
  - '(I) killed the man' `[killed,OBJ,man]` ;
  - 'The air is compressed (by subject)' `[compressed,OBJ,air]` ;
- ATTR(ibutive):
  - 'the smaller wheel' `[wheel,ATTR,smaller]`;
- PRED(icate):
  - 'the element is uranium'

`[element,PRED,uranium]`;

- MOD(ifier):
  - 'very green'
    `[green,MOD,very]`;
- QUANT(ifier):
  - 'four wheels'
    `[wheels,QUANT,four]`;
- ... .

We did not apply any lemmatisation (or stemming) to the words in the triplets.

To limit the time needed to parse all 39,802 documents, we decided to introduce a maximal time limit for the parser (1800 seconds per parse). With this procedure two topic documents failed to parse, as well as a very small fraction of documents returned in the retrieval process (0.0025%). Though this may not seem much, it does mean that every time the parser failed, absolutely no triplets were generated for that portion of the text, which makes it invisible for the retrieval system in the reranking experiment. Numerically, however, these missing triplets are only a fraction of the corpus of some 32 million triplets that were generated. In total, it took a week to parse the topics and the documents in the result sets on a cluster of single core PCs, most of which had no more than 1 Gbyte of internal memory.

3.3.3 Query and indexing

Triplets from both the topic and result set documents were transformed into a single string using a perl script. For example, `[fact,ATTR,well-known]` and `[system,SUBJ,performs]` were transformed into `factattrwell_known` and `systemsubjperforms`, respectively. These strings then served as index and query terms for direct matching ('Bag of Triplets' matching). We constructed 500 separate indices (one per topic) using the BuildIndex function of Lemur with the indri IndexType. Each index contained the strings of those documents that were retrieved for that topic in the baseline run. For each of the 500 topic queries, batch retrieval was then performed on its specific index using the TF-IDF ranking algorithm.

Since we performed a second retrieval step, we take the risk of not re-retrieving a portion of the documents retrieved with the BOW retrieval. On average, we retrieved 90.1% of the 100 documents per topic. We identified the missing documents using a python script that compared the baseline result list with the reranking result list and added these missing documents to the end of the reranking result list in the relative order in which they had been found in the baseline.

The 498 successful[14] individual retrieval result sets of the reranking experiment were compared with the results from the baseline experiment using a python script in order to calculate the Precision, Recall and MAP measures. We also calculated the rank of the first relevant document per query.

[14]As mentioned above, two topic parses failed and therefore we could not compare the retrieval sets.

4. RESULTS

4.1 Baseline retrieval results

During the baseline experiment we retrieved a total of 645 relevant documents in the CLEF-IP corpus for the 500 topic documents. We achieved a score of 0.22 for recall and a MAP score of 0.054.

4.2 Reranking retrieval results

The reranking system performed significantly worse than the baseline system: The MAP score dropped from 0.054 for the baseline system to 0.045 for the reranking system. ($p < 0.001$ according to the Wilcoxon Signed Ranks Test).

Of the 645 relevant documents that were retrieved in both experiments, 8 had the same ranking in the baseline as in the reranking result set (1.3%). In 537 cases, the relevant document had a higher ranking in the baseline approach (83.3%) and only in 100 cases did the reranking approach produce better rankings for the documents (15.5%). On average, the documents either dropped 40 ranks in the reranking result set or rose 18 ranks compared to the baseline rankings.

4.3 Parser evaluation

The outcome of the retrieval process is highly dependent on the quality and quantity of the generated triplets. We therefore evaluated the accuracy of our parser on a small test set of 14 sentences (656 words) taken randomly from the claims sections. To create this test set two of the authors independently created dependency triplets for different parts of the test set. There was an overlap of 5 sentences, each of around 40 words, which was used to calculate inter-annotator agreement for the test set. The inter-annotator agreement was 74%[15], indicating substantial agreement on annotation. The language typically used in claims sections ('legalese') has – apart from other particularities – a lot of syntactic ambiguities and therefore it is not surprising that the biggest differences in manual annotations could be attributed to different interpretations of coordinations for the SUBJ relations and of PP attachments. The following sentence is an example of the first difficulty:

> 'The device claimed in claim 1 consists of .... [15 words] and uses 5 volt.'

As these dependencies can be stretched quite long (15 words in between), it is very difficult, even for a human, to see which word should be connected to the second verb. An example of the second problem can be seen in the following example:

> 'The mapper is adapted to divide a stream of bits from the encoder into at least a first period by the rightful application of ... ' .

This is a well-known problem for any parser, usually demonstrated with the famous 'I saw the man with the telescope'-example, but because of its frequency it becomes even more

[15]The percentage of triplets that were identical and correct in both the annotation sets

problematic in parsing patent language. There was high agreement on triplets containing the ATTR and OBJ relations (94% and 83% respectively). Differences in annotation were resolved by discussion, and the resulting set of annotations was used to evaluate the parser.

Parsing accuracy was rather low: On the test set of 14 sentences the parser achieved 0.37 in precision (the number of correctly generated triplets divided by the number of all generated triplets) and 0.31 in recall (the number of correctly generated triplets divided by the number of all correct triplets), so accuracy[16] rated 0.34. Thus the parser generated a lot of incorrect triplets, while it also generated too few triplets (333 generated versus 416 manually annotated). The latter is a consequence of gaps in the lexical and syntactic coverage of this version of the parser. We tested the lexical coverage and 98% of the words in the corpus featured in the parser lexicon[17] or were robustly recognized (see infra). The only words that the parser could not recognize were chemical formulae. It is difficult to say something about the syntactic coverage of the parser for this type of language: We previously tested the same parser on a general language regression test set of about 300 short sentences. On this set the parser achieved 0.87 accuracy. It is unlikely that the grammatical constructions used in patent texts are so different from those used in general language that the parser would perform so badly on this kind of text. More likely, the low accuracy is a consequence of some gaps in the grammatical coverage and the difference in language use that we observe in patent texts.

Looking at the faulty triplets, we noticed that quite often these were caused by lexical ambiguity or incomplete POS information in the lexicon: when a word is taken to be a verb, while it is in fact a noun or an adjective, this will have a profound effect on all the triplets in which a word connects with this verb. For example, during the analysis of the parser triplets, we noticed that the quality of the triplets containing SUBJ or OBJ relators was exceptionally bad (0.42 and 0.22 accuracy respectively). Analysis of the sentences showed that the erroneous interpretation of 'said' (as in 'the second screw in said device') as a verb instead of an adjective created at least four faulty triplets per occurrence, e.g. `[screw,SUBJ,said]`, `[said,OBJ,device]`, `[said,PREPin,second]`, ...

In order to be able to deal with all sorts of text, our parser is equipped with a few robust rules, which can robustly recognise words that are not in the lexicon and give them a part of speech (for example any word ending in -ly that can not be found in the lexicon, will be recognised as an adverb), or assign a part of speech to a word that is different from what is mentioned in the lexicon (for example, the fact that the verb 'run' can become a noun in 'The first run of the cycle went fine.' is covered in the grammar rules). On the one hand, such robust rules improve the recall of the parser as some of the terminology in the patent texts is not included in the lexicon and must therefore be recognised by other means. Furthermore, the language use in patents is quite different from general language use: The different POS possibilities of the word 'said' is a clear example of that. On the other hand, such robust rules must be used with caution: If used too liberally they can pose a risk for precision, because they make the parser more likely to generate faulty triplets. If any noun were allowed to be a verb and any adjective a noun or a verb, even a simple phrase like 'a good book shop' would have at least four parses with the following interpretations: 'a good shop for books', 'a good book that shops',' goods that shop for books', 'goods that book a shop'. These would render different triplets and without any extra information it would be impossible for the parser to identify the correct parse.

At this moment we are experimenting with a hybrid version of the parser in which the parsing process is guided by frequency information of good[18] dependency triplets in patent texts. This way the robustness of the parser remains intact, but the proliferation of faulty triplets is kept to a minimum.

5. DISCUSSION

In this section some analysis is done trying to identify the reasons behind the bad reranking performance of our second step in the patent retrieval task. There are three reasons why the MAP score is so much lower in the reranking experiment compared to the baseline.

First of all, it seems that Dependency Triplets – in their current form – are too detailed to be used as index terms. In the reranking experiment an average of 90.1% documents was returned. This means that for almost 10% of the documents there was no overlap between the triplets in the patent topic and the documents returned in the first retrieval step. While this specificity is problematic for the retrieval results, it is also the greatest strength of the linguistically-based system. We should find the correct balance between detailed information and more general index terms by adding extensive lexical normalisation to our system. If the triplets contain lemmas instead of word forms, a great deal of the morphological variation will disappear and overlap should increase. As we use a parser with an extensive lexicon to generate the dependency triplets, lemmatisation is not a very difficult step to implement. Another strategy would be to stem all the word forms in the dependency triplets before they are used in the retrieval process. This would be less effective than using lemmatisation: lemmatisation is more selective than stemming since a single stem can be the basis of more than one lemma; Furthermore, cropping the word forms to their stems would make the dependency triplets less informative when they are used to guide the parsing process of the hybrid parser.

A second reason why this experiment yielded negative results is the gaps in the triplet coverage of the documents. As mentioned above, the parser was not able to parse all

[16]This is the F1-measure, calculated from the precision and recall achieved by the parser.

[17]The fact that a word is found in the lexicon (lexical coverage) does not necessarily mean that the lexical information is complete and accurate for all uses and contexts. For example, if the word 'chair' is known in the lexicon only as a noun in the sentence 'He needs people to chair the first session.' where 'chair' is a verb, the parser will fail to produce the correct parse.

[18]This means reliable triplets, irrespective of the context in which they were found.

the documents completely: two topic documents completely failed to parse and in about 2% percent of the retrieved documents, the parser failed on part of the text, thus creating holes in the triplet coverage of that document, which may be crucial to the retrieval process. It is clear that we need to add another pre-processing step to our system to make sure that unparsable (large) sections are split up into smaller units that the parser can manage.[19] We estimate that the parser should have generated around 40 million triplets, instead of the 32 million that have been produced in this experiment. Triplet coverage should also improve when the grammatical coverage improves, more specifically for those structures that are typical for patent texts.

The final and probably most important reason for these low scores is the bad quality of the generated triplets. As our system depends on exact matching of detailed (and consequently low frequency) terms, lowering the frequencies with which the terms (triplets) occur by assigning some occurrences to faulty triplets has a very harmful effect on the retrieval process.

As mentioned above the language used in the claims section is very difficult to parse, even for humans, and it is quite possible that using the language from the abstract or description fields would have yielded better results for this experiment. The claims section is, however, a very important part of the patent text and our parser must be able to parse the language correctly. We are now working on a hybrid parser that uses information about triplet frequency to guide the parsing process. By supplying it with a set of correct triplets that are typical for the language used in claims, the parser should be able to deal with lexical ambiguities. Better syntactic coverage will also improve the parser's performance.

6. CONCLUSIONS

In this paper, we described a reranking experiment following our participation in the CLEF-IP 2009 track. We explored whether using syntactic structures represented by means of dependency triplets as index terms would lead to improvements in the reranking of the relevant documents that were found in the baseline run for the CLEF-IP track. Our experiment illustrated the difficulties of generating good quality triplets for retrieval purposes. We were not able to improve the ranking in the second step. On the contrary, the MAP scores were significantly lower for the reranking experiment. This was caused by the following factors: a) the overall quality of the triplets was low; b) there were gaps in the triplet coverage of the documents due to parse failures; c) there was not enough overlap between topic and corpus triplets because the triplets are too detailed in their current form.

For future work, we need to improve the parser accuracy both for lexical and syntactic ambiguity. We believe that using a hybrid parser with triplet frequency information will have a significant effect on the quality of the generated triplets. We also need to use lemmas instead of word forms in our dependency triplets in order to improve overlap between the topic and corpus documents.

When these improvements have been implemented in the parser, this experiment should be repeated in order to find conclusive evidence whether or not dependency triplets can improve the reranking of relevant documents found by a BOW approach in patent retrieval. If the results are equally poor, we will have to revisit our arguments that predict that triplets are conducive to this task.

[19]For example: the entire claims section consists of one, immense sentence. By splitting this sentence up into smaller, more manageable clauses, we could improve parsing speed and produce more triplets for this section.

7. ACKNOWLEDGMENTS

The TM4IP project is being funded by Matrixware.

8. REFERENCES

[1] M. A. Alonso, J. V. Ferro, and V. M. Darriba. On the Usefulness of Extracting Syntactic Dependencies for Text Indexing. In *AICS '02: Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science*, pages 3–11, London, UK, 2002. Springer-Verlag.

[2] A. Arampatzis, T. van der Weide, C. Koster, and P. van Bommel. *Encyclopedia of Library and Information Science*, chapter Linguistically-motivated information retrieval. Marcel Dekker, New York, 2000.

[3] Atsushi Fujii, Masao Utiyama, Mikio Yamamoto and Takehito Utsuro. Producing a Test Collection for Patent Machine Translation in the Seventh NTCIR Workshop. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[4] H. BaoQuoc, T. B. T. Dong, J. Chevallet, and M. Bruandet. A structured indexing model based on noun phrases. In *RIVF*, pages 81–89, 2006.

[5] D. Bonino, A. Ciaramella, and F. Corno. Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics. *World Patent Information*, June 2009.

[6] T. Brants. Natural Language Processing in Information Retrieval. In *Proceedings of the 14th Meeting of Computational Linguistics in the Netherlands 2003*, 2004.

[7] L. Chen, N. Tokuda, and H. Adachi. A patent document retrieval system addressing both semantic and syntactic properties. In *Proceedings of the ACL-2003 workshop on Patent corpus processing*, pages 1–6, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[8] H. Cui, R. Sun, K. Li, M.-Y. Kan, and T.-s. Chua. Question answering passage retrieval using dependency relations. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 400–407, 2005.

[9] E. Escorsa, M. Giereth, Y. Kompatsiaris, S. Papadopoulos, E. Pianta, G. Piella, I. Puhlmann, G. Rao, M. Rotard, P. Schoester, L. Serafini, and V. Zervaki. Towards content-oriented patent document processing. *World Patent Information*, 30(1):21–33, 2008.

[10] J. Fagan. *Experiments in Automatic Phrase Indexing*

For Document Retrieval: A comparison of Syntactic and Non-Syntactic Methods. PhD thesis, Cornell University, 1987.

[11] J. Gao, J.-Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, New York, NY, USA, 2004. ACM.

[12] E. Graf and L. Azzopardi. A methodology for building a test collection for prior art search. In *Proceedings of the 2nd International Workshop on Evaluating Information Access (EVIA)*, pages 60–71.

[13] C. S.-G. Khoo. The Use of Relation Matching in Information Retrieval. *LIBRES: Library and Information Science Research Electronic Journal*, 7(2), 1997.

[14] C. Koster and J. Beney. Phrase-Based Document Categorization Revisited. 2009.

[15] C. Koster, N. Oostdijk, S. Verberne, and E. Dhondt. Challenges in Professional Search with PHASAR. 2009.

[16] W. Kraaij and R. Pohlmann. Comparing the Effect of Syntactic vs. Statistical Phrase Indexing Strategies for Dutch. In *ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 605–617, London, UK, 1998. Springer-Verlag.

[17] H. Mase, T. Matsubayashi, Y. Ogawa, M. Iwayama, and T. Oshio. Proposal of two-stage patent retrieval method considering the claim structure. In *ACM Transactions on Asian Language Information Processing (TALIP)*, volume 4, pages 190–206, 2005.

[18] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479, New York, NY, USA, 2005. ACM.

[19] M.-F. Moens. *Automatic Indexing and Abstracting of Document Texts*, volume Vol.6 of *The Kluwer International Series on Information Retrieval*. 2005.

[20] G. Roda, J. Tait, F. Piroi, and V. Zenz. CLEF-IP 2009: Retrieval experiments in the Intellectual Property domain. 2009.

[21] S. Sheremetyeva. Towards Designing Natural Language Interfaces. In *Proceedings of the 4th International Conference "Computational Linguistics and Intelligent Text Processing"*, 2003.

[22] A. Shinmori, M. Okumura, Y. Marukawa, and M. Iwayama. Patent claim processing for readability: structure analysis and term explanation. In *Proceedings of the ACL-2003 workshop on Patent corpus processing*, pages 56–65, 2003.

[23] A. F. Smeaton and C. J. van Rijsbergen. Experiments on incorporating syntactic processing of user queries into a document retrieval strategy. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 31–51, New York, NY, USA, 1988. ACM.

[24] F. Song and W. B. Croft. A general language model for information retrieval. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321, New York, NY, USA, 1999. ACM.

[25] Y.-I. Song, K.-S. Han, S.-B. Kim, S.-Y. Park, and H.-C. Rim. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems*, 31(3):265–286, 2008.

[26] T. Strzalkowski, J. Carballo, and M. Marinescu. Natural language information retrieval: TREC-3 report. Technical report, In The Third Text Retrieval Conference (TREC 3), 1994.

[27] Y. Tseng and Y. Wu. A study of search tactics for patentability search: a case study on patent engineers. In *Proceeding of the 1st ACM workshop on Patent information retrieval*, pages 33–36, 2008.

[28] S. Verberne, L. Boves, N. Oostdijk, and P.-A. Coppen. What is not in the Bag of Words for Why-QA? 2009. To appear in *Computational Linguistics*.

[29] C. Zhai. Fast statistical Parsing of Noun Phrases for document Indexing. In *Proceedings of the fifth conference on Applied natural language processing*, pages 312–319, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

Knowledge-driven Information Retrieval for Natural History

Marieke van Erp
Vrije Universiteit
De Boelelaan 1081a
Amsterdam, The Netherlands
marieke@cs.vu.nl

Steve Hunt
TiCC/Tilburg University
Warandelaan 2
Tilburg, The Netherlands
S.J.Hunt@uvt.nl

ABSTRACT

We present an information retrieval system that utilises domain knowledge from an ontology and external resources to aid retrieval of records from two natural history databases. The domain knowledge is inserted on three different levels, namely (1) to aid query formulation, (2) to expand queries with relevant synonyms and (3) to rank results. Query interpretation alone facilitates a rise in mean average precision scores of 7.59% for the first database, for the second database the result of query interpretation is not significantly better. Query expansion together with query interpretation boosts recall from 31.67% to 85.85% for our first database. For the second database, query interpretation and query expansion account for an increase in recall from 46.83% to 87.22%. In the most lenient query mode (in which records are retrieved that match any of the query terms), ranking can remedy some of the negative effects induced by the retrieval of large numbers of irrelevant records.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation; H.2.8 [**Database Applications**]: Scientific databases

General Terms

biodiversity informatics, information retrieval

1. INTRODUCTION

Biosystematics is concerned with describing life forms. Biodiversity research is concerned with describing the interaction of different life forms within ecosystems. Both research domains depend heavily on specimen collections such as those found at natural history museums. Such specimen collections consist of two parts: (1) the animal specimens themselves and (2) the textual information that describes where, when, and under what circumstances the specimen was collected.

Access to the textual information accompanying each specimen is often impaired by lack of digitisation or unawareness of the opportunities digitisation brings about. We report work that is aimed at improving access for researchers from the Dutch National Museum of Natural History Naturalis[1] to the textual information that accompanies the objects in their collection. Currently, access to the database is through standard database systems such as Microsoft Access and MySQL. However, these database systems require that the user has extensive knowledge of the database structure and changes in the natural history domain. To alleviate this, we present a system in which retrieval of more relevant records is enhanced through inserting domain knowledge in the retrieval process. The extra knowledge works on three steps in the retrieval process: (1) query interpretation to facilitate complex queries that cannot be dealt with through simple and/or search, (2) query expansion to remedy the negative effects of the use of synonyms, and (3) result ranking to present users with more relevant results first.

In this work we present the following four contributions.

1. a retrieval system for natural history
2. insertion of domain knowledge in all stages of retrieval
3. active use of the database structure to aid ranking
4. experiments on two databases from the natural history that show that domain knowledge improves retrieval

This paper is organised as follows. In Section 2, related work on domain specific retrieval is discussed. In Section 3, the resources used for our experiments are described. In Section 4, our knowledge-driven retrieval system called Mira is described. In Section 5, we describe our experiments and results. In Section 6 the results are discussed which is followed by our conclusions in Section 7.

2. RELATED WORK

A substantial body of work on query interpretation and reformulation exists. Tata and Lohman [18], for example, developed a system that translates keyword queries into SQL queries to query databases. Their approach utilises a parser to match query terms to database schema elements and a set of rules to generate and rank possible query trees. A similar approach is taken in [22] in order to construct SPARQL

[1] http://www.naturalis.nl

queries from keyword queries. Background knowledge from ontologies are used to translate keyword queries to description logic conjunctive queries in [19]. In [19], query terms are matched to knowledge base entities, after which connections between the selected knowledge base entities are identified and listed in a graph. This graph is translated to the final query.

In many retrieval and search tasks, one has to deal with the fact that there there are synonyms present in the data or a simple keyword search may be ambiguous. To remedy this, query expansion is a popular topic in information retrieval. Query expansion to aid retrieval was first introduced in [17]. In this work, a clustering technique was used to discover related terms based on co-occurrence in documents to expand queries with semantically similar terms. In [9], query expansion is performed through also searching for matches with synonyms for a query term found in WordNet. In [15], the authors use Wikipedia instead of WordNet to identify synonymous terms to expand queries with. In addition to this, results are clustered by topics which are browsable by users, thus providing an extra means to filter out possibly negative results.

Domain-specific resources to identify terms with which to expand queries with are used by [3], [5] and [14] and were also popular in the TREC Genomics track that ran from 2003 to 2007 [11]. The authors of [3], use domain specific knowledge bases (as well as WordNet) to search for synonymous, as well as hypernymous and hyponymous, terms to expand queries with. In [5], queries are automatically expanded with terms taken from domain specific databases. The authors achieve mixed results: some of the databases they used proved to be more useful than others. The most similar to our system is the system described in [14] to retrieve relevant abstracts from medical literature. As in MIRA, domain knowledge is utilised in various stages of the search process. The system in [14] first analyses the queries to inform the system of what type of information is often searched for. At a later stage domain knowledge is used to re-rank the results.

Our approach differs from the approaches discussed above in two ways. First, we introduce domain knowledge to all stages of retrieval (query analysis, query expansion and result ranking). Second, we utilise the structure of the database to aid the retrieval process.

3. RESOURCES

We test our approaches on two databases from the natural history domain, these are described in Subsection 3.1. In order to test the system, researchers provided us with 100 test queries for each of the databases, the test queries are described in Subsection 3.2. The domain knowledge used in the system comes from external resources, which are described in Subsection 3.3.

3.1 Databases

The databases used for this research are created at the natural history museum. They are transcribed from field notes (written up by biologists whilst collecting specimens in the field), registers (records describing when and what specimens entered the museum and what specimens left, for example, on loan to other institutes), and taxonomies (accepted lists of animal classifications).

Reptiles and Amphibians

The Reptiles and Amphibians database is a resource compiled from a manually created database containing 16,870 records and an automatically populated database containing 39,688 records. Each record describes where, when and under what circumstances a reptile or amphibian specimen in the Naturalis collection was found and how it is preserved. The manually created database was compiled by researchers at the institution. It contains 37 columns. The automatically populated database was created by automatically segmenting and labelling the field notes and registers (this process is described in [13]).

The database is mostly composed in Dutch and English, but also contains some information in German and Portuguese.

Birds

The database describing the birds collection that is used in this work is compiled from three smaller databases describing birds specimens at Naturalis. The first two databases were created within the *Building the databases of life*[2] project at Naturalis to convert collection information from paper resources to digital resources. The first database contains 117,649 records describing the passerines (songbirds) that are preserved as study skins in the collection. The second database contains 68,341 records describing the non-passerines (non-songbirds). The third database was created later by researchers and contains 34,028 records that describe specimens in the collection that were not included in the previous two databases (such as the mounted passerine specimens). The combined database (henceforth referred to as *Birds database*) contains 220,018 records in total. The information is presented in 32 columns. As with the reptiles and amphibians database, it contains entries in Dutch and English. Indonesian, German, French and Portuguese entries are also found.

3.2 Queries

External researchers often request access to Naturalis' extensive specimen collection or to the meta-data that is found in the databases describing the collections. As the databases are not publicly available, these questions are usually directed to the collection managers at Naturalis. To test the MIRA system, collection managers have saved these questions they received regarding the reptiles and amphibians and bird collections. These queries give a good idea of the type of information researchers are looking for.

Both the bird and reptiles and amphibian questions were extracted from longer (often email) messages. The questions have been summarised into only the information request and not the introduction for why the information is requested.

For each of the queries the relevant records in the databases were identified manually to create a gold standard.

[2]NWO Groot project number: 175.010.2003.010

Reptile and Amphibian Queries
The 100 reptile and amphibians queries were gathered from requests to the reptile and amphibian collection managers and researchers at Naturalis that were received between September 2003 and December 2008.

Some example queries are:

- What type specimens of New Guinee skink do you have in your collection?
- Do you have male specimens of Hypsilurus godeffroyi?
- Are there Dipsas species other than D. catesbyi and D. variegata from the Guianas and Venezuela in the collection?
- How many species of Rana palmipes as defined by Spix in 1824 are in the collection?

12% of the questions enquire after a genus, 86% after a genus and a species and in 41% the request poses a restriction on the geographical location of where the specimen was collected. Additionally, in 15% of the questions a registration number is given, which should make it easier to retrieve correct database record but as registration numbers are not unique this is not always the case.

For 16 queries no relevant records were present in the database, for the remaining 84 queries the number of returned records varies greatly. For example, for 21 queries only 1 relevant result is present in the database whereas there are 4 queries for which over 500 relevant results are present in the database.

Bird Queries
The 100 queries for the birds experiments were gathered from requests to the bird collection managers sent between 1992 and 2006.

Some example queries are:

- Are there any specimens Nipponia nippon in the Naturalis collection?
- Are there any striped crakes (Aenigmatolimnas marginalis) from Africa collected by Andersson in 1867 in the collection?
- Is there a juvenile female specimen of Hypotaenidia celebensis (now Gallirallus torquatus celebensis) in the collection?
- Are there any skins of Leclancher's Bunting (P. leclancherii) in the collection?

5% of the questions inquire after a genus, 93% after a genus and a species and in 6% the request poses a restriction on the geographical location where the specimen was collected. Contrary to the reptiles and amphibians questions, there are no mentions of registration numbers, this is due to the fact that in this domain it is not customary to mention the registration number, whereas in the reptiles and amphibians domain it is. The requests for information on the birds collection do specify what type of material is needed for the research, as 30 % the question mentions skin or skeleton. The reason that this is not relevant for most herpetological research is that those specimens are most often kept in alcohol, whereas birds are mostly kept partly or completely dried and mounted.

As with the reptiles and amphibians queries, for 16 queries no relevant records were available in the database and there is a great variety in the number of relevant records for the remaining 84 queries.

3.3 External Resources
The query interpretation and query expansion modules in Mira utilise taxonomic and geographic information about the reptiles and amphibians and birds domains from external resources. Below, we describe two taxonomic resources (one for amphibians and one for reptiles). For the birds domain there was no high quality digital taxonomic resource freely available. After the taxonomic resources, the geographical resource that was used is described.

Taxonomic Resources
For the amphibians, the Frost taxonomy is used, as published online [8]. The version used in this work (version 5.3) contains descriptions of 6,433 amphibian specimens with references to the literature and synonyms.

For the reptiles, the TIGR Reptile Database [20] is used. It is compiled from books, checklists, monographs, journals, and other peer-reviewed publications from the domain of reptile taxonomy. It is currently maintained by the Systematics working group of the German Herpetological Society (DGHT). It lists all species and their position in the taxonomy. 8,600 reptile species are described.

GeoNames
GeoNames[3] is an aggregated geographical data base that is available through a Creative Commons attribution license and accessible through various Web services.

The GeoNames database is compiled from a collection of smaller geographic resources. In June 2009, GeoNames contained over eight million geographical names, of which 6.5 million unique entities. It is an attractive resource to pair the Naturalis data with as it contains alternative names for geographic entities in numerous languages.

4. MIRA ARCHITECTURE
In this section, the Mira system setup is presented. An overview of the system is presented in Figure 1. The domain knowledge comes from taxonomic resources, geographic resources, a domain ontology, domain specific rules and analysis of typical queries in the domain. Below, each of the Mira modules is described.

4.1 Query Interpretation
Most of the queries in the test sets require more precise formulation than simply and/or queries. Consider for example

[3] `http://www.geonames.org`, Last queried 15 July, 2009

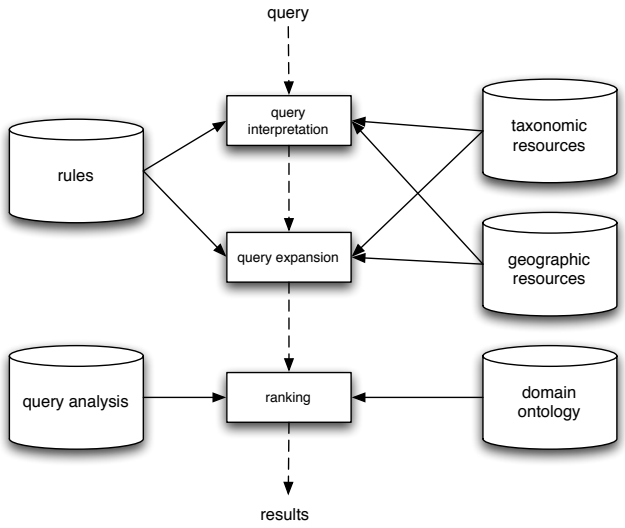


Figure 1: Overview of the MIRA system.

the query *Are there Dipsas species other than D. catesbyi and D. variegata from the Guianas and Venezuela in the collection?*. Here, the user is looking for database records that describe specimens of genus Dipsas, but not those records of species Dipsas catesbyi and Dipsas variegata. The second constraint is that the user wants the relevant records about specimens collected in the Guianas or Venezuela.

To be able to handle such complex queries, we devised a query language that can encode that for part of the query any query term should match and for part of the query all query terms should match. The query language can also exclude terms on the basis of a negation. The query terms that we extract from the example query are: *dipsas*, *-catesbyi*, *-variegata*, *guianas* and *venezuela*. To express that specimens of genus *Dipsas* found in the *Guianas* or in *Venezuela* are to be retrieved, the query is rewritten to *all(dipsas,any(guianas,venezuela))*. To exclude the records on specimens of species *catesbyi* and *variegata* the query is written as *all(dipsas,-catesbyi,-variegata),any(guianas,venezuela))*.

Users can be taught this query format, but due to the availability of taxonomic resources MIRA can also automatically translate basic query term enumerations such as *dipsas*, *-catesbyi*, *-variegata*, *guianas* and *venezuela* into the desired complex query for the reptiles and amphibians. In order to do so, it looks up each query term in the taxonomic and geographic resources to classify it as either a genus, species or geographic name. The module can also recognise registration numbers as terms that contain two or three capital letters and 3 to 6 numbers. After each term is classified, the module builds up the query according to rules that restrict possible combinations of types of terms.

Initially, every query is translated as *all(...,...)*, denoting that all terms between the brackets should occur in a document for it to be retrieved, as one wants as many terms as possible to match. Then, the system tries to classify each term as either indicating a genus, species or geographic or registration number. If two terms of the same type are identified within the same query this could indicate a synonym and this part of the query is then relaxed to facilitate for records to be retrieved that include either one of the synonymous terms. The synonymous terms are thus embedded in an *any(...,...)* pair. Cases in which two genus/species pairs are encountered are translated to *any(all(genus$_1$,species$_1$),all (genus$_2$,species$_2$))* and are expanded if more than two pairs are found. The automatic translation module is checked against a gold-standard of manual rewriting of each query. For the reptiles and amphibians, it translates 77% of the questions correctly. The cause for translation module failure is, in all cases, due to a term not matching in the resource. For the birds, no taxonomic resource was freely available to run the automatic query interpretation module, therefore in the experiments presented in Section 5, the manually rewritten queries are used.

4.2 Query expansion

The query expansion modules in MIRA are aimed at increasing the recall by providing additional keywords or to remedy the influence of language variation on the retrieval of relevant results.

Taxonomic term expansion

Although the foundation for the zoological taxonomy that was laid by Linnaeus has not changed, discoveries through, for example DNA research, have caused many species classifications to be revisited. Therefore, accepted taxonomic lists that describe the classification of a taxonomic class (such as reptiles), contain many synonyms and outdated names for each taxon. For example, if one wants to retrieve all snakes present in the collection, one could query for all records describing a specimen of suborder 'Serpentes'. Unfortunately, the suborder 'Serpentes' is also known as 'Ophidiae'. An additional problem with this query is that the reptiles and amphibians database does not contain a suborder column (although sometimes the suborder value is entered in the order field), hence in order to retrieve all snakes in the collection one would have to query the database for all 18 snake families, which each may be known by synonyms as well. To relieve users from needing to formulate a query that contains each of the 18 snake families with their possible synonyms, MIRA applies a knowledge-based query expansion approach that expands query terms with their taxonomic synonyms.

Geographic term expansion

Similar to the taxonomic term expansion, but slightly different in operation is the enrichment of MIRA with a geographic resource. If we reconsider the example given in Subsection 4.1, *Are there Dipsas species other than D. catesbyi and D. variegata from the Guianas and Venezuela in the collection?*, we notice that *the Guianas* does not denote one country, instead it denotes Guyana (formerly British Guiana), Suriname (formerly Dutch Guiana) and French Guiana. Furthermore, for each of these names, there are alternate spellings and the fact that our database is made up in several languages may also impair relevant records from being retrieved. Fortunately, GeoNames contains many of the synonyms to automatically expand our query with.

Several flavours of a geographical expansion module were

investigated, such as in addition to expanding to synonymous terms (for example in different languages), expansion to hypernyms or hyponyms following the idea of [21]. In a hypernym module, if the query contained the term 'Nebraska', the query was expanded to 'United States of America' to remedy the negative influence of missing values in the 'province/state' column. Although hypernym and hyponym expansion are popular approaches that work for other systems (see [16] for an overview) it did not aid object retrieval for the herpetological and birds collections in these experiments. Therefore the geographical expansion was limited to expanding only to synonymous terms and location names in different languages.

4.3 Ranking

In order to present the user with the more relevant records first, two ranking methods were investigated.

RecordRank

RecordRank is a simplified version of the basic PageRank algorithm developed by the founders of Google in 1998 to rank results by relevancy [4]. The main assumption behind PageRank is that some webpages are more authoritative than others and those should rank higher than pages that are deemed less authoritative. The idea to rank the MIRA retrieval results by some measure of authority is given by the hypothesis that researchers might pose more questions about the specimens or species Naturalis is known for (e.g., the reptiles and amphibians collections contain many specimens from the Amazon, therefore researchers might ask more about that part of the collection than about specimens collected in Africa as there are fewer of those).

Authority in PageRank is measured by the number of incoming links to a page. Also, links from pages with a higher PageRank are considered more important than links from pages with a lower PageRank.

The PageRank algorithm has sparked interest in applications other than search engines as ranking results for entity relation graphs [6] and Word Sense Disambiguation [1]. Similar to the aim in MIRA, the PageRank algorithm has also been translated to a relational database setting in [2]. In this work, databases are translated to modelled graphs in which objects are nodes and their semantic connections the edges. Although the databases used for MIRA were originally flat, a domain ontology that was developed for the natural history domain can enrich the databases with the necessary structure to consider them as a relational data resource.

Once the databases are relational, it would seem straightforward to follow [2] and convert the database to a graph and apply their ObjectRank algorithm to it. Due to the fact that all relations in the ontology are bi-directional the approach is even simpler because the ObjectRank scores can be approximated by taking a shortcut of ranking objects by degree. The notion of degree comes from Graph Theory and denotes the number of edges (relations) linked to a node (object) [7]. To compute the rank-score of an object, the number of relations is counted and objects with a high number of relations get a higher score and thus higher rank.

In order to go from a ranking of objects in the domain to a ranking of records in the database the scores of all objects that occur in a database record are added up and normalised over the number of objects present in the database record (as database cells can be empty). For every database record the scores of every value are added up resulting in a RecordRank score by which the database records can be ranked.

A drawback of RecordRank is that for broader queries in a smaller domain the same set of database entries is always ranked on top. It may therefore be more useful to present a ranking of importance relative to a query. This idea was explored in [10], who presents a topic-sensitive PageRank approach. The idea of only computing the rank over the retrieved result is also used in the HITS algorithm, another link analysis algorithm that is used to rank web pages according to authority [12]. In [10]'s approach, a set of topic-specific PageRank vectors is computed only from pages relevant to the query, which are then used to retrieve results for a query on a particular subject. Since the natural history databases provide a smaller domain which cannot be easily broken up in more subdomains, the MIRA query-sensitive RecordRank module does not use precomputed vectors. Instead, for each query the RecordRank scores are computed at run-time, but only for the retrieved results. We distinguish the two flavours of RecordRank as Global RecordRank, in which database records are ranked by authority regardless of the query, and Local RecordRank, in which database records are ranked after records are retrieved.

Column order by importance

Analysis of the queries has shown that queries do not usually pertain to information in some of the longer database columns such as special remarks. Hence, when giving each column equal importance a query such as *Bufo marinus* will return results such as:

> *RMNH 34003* ***Bufo marinus*** *Lely Range, airstrip, distr. Marowijne, Surinam, 11-05-1975, 15.50h, on airstrip, near tall forest, 650m, l + d. X.X. Xxxxxxxx. RMNH 34003*

as well as:

> *RMNH 20761 TANK NO Slide 1980-10- 37 (fell) Paleosuchus trigonatus 1 ex. km 110, 19-09-1980, 20.45 h, in swamp, flooded part of forest with many dead trees and low bushes, near jeep trail through tall forest, 100 m. length 1.445 m, skin and carcass to create skeleton. Stomach contents kept separately: crab +* ***Bufo marinus*** *+ grit. Observed this specimen already on 16-09-1980 (see p.89).*

After analysis of the queries it was clear that a large majority of the queries pertain to the request for information from the genus and species columns and never from the special remarks column in which one might find information on a specimen's stomach contents. Records with matches found in these columns, as well as in the registration number column are thus presented before records with matches found in other columns.

5. EXPERIMENTS AND RESULTS

In this section, the results of the experiments of the retrieval of records from the reptiles and amphibians and birds database with MIRA are presented. Only the first 5000 results returned for each query are evaluated using the evaluation script used in the Text REtrieval Conferences (TREC)[4]. In each of the tables presented below, the bold face results are significant with respect to the baseline results that the module is compared to. All significance scores are computed at the p=0.05 level using a paired t-test. The ALL query mode denotes a simple keyword search in which only records should be retrieved in which all query terms match. The ANY query mode is another simple query mode in which records should be retrieved in which any of the query terms match. The interpreted query mode (as described in Subsection 4.1) is denoted by COMPLEX in the tables.

The precision, recall and mean average precision (MAP) for the interpretation and expansion modules are presented in Table 1. As the results in Table 1 show, the ALL query mode benefits more than the ANY query mode of the query expansion. This is due to the fact that the ANY query mode already achieves high recall, simply because it retrieves records in which at least one of the query terms match. Separately, the expansion modules perform best (denoted by TAXEXP for taxonomic expansion and GEOEXP for geographic expansion). When combined, and thus when they expand both the geographic and the taxonomic queries (TAXGEOEXP), the achieved results are mixed. For the ALL query mode, the precision does not deteriorate significantly (whereas it does for the separate expansion modules), but recall does not improve as much as expected, therefore this module is not further investigated. This is probably due to an explosion of expanded terms for each query term and the subsequent retrieval of too many records.

The experiments carried out with the query interpretation module are found in the lower part of Table 1. The precision and mean average precision scores for the interpreted query mode are significantly higher than for the simple query modes. On its own, the COMPLEX query mode improves the mean average precision with 5.83% over the ALL query mode, and with 7.59% for the ANY query mode. Together with the query expansion modules, the COMPLEX query mode helps improve the scores even more, in particular the geographic expansion module. The difference in recall between the unexpanded ALL query mode experiments and the geographically expanded COMPLEX query mode experiments is even more than 50% (from 31.67% to 85.85%). Also the ALL query mode benefits from query expansion.

In Table 2 the mean average precision scores for the ranking modules are presented. Our assumption that the RecordRank modules would aid performance because the more authoritative records are presented first proved wrong. For the unexpanded queries, the mean average precision improves, but not significantly. For the expanded queries, the RecordRank modules even harm performance.

Due to the precise manner of querying provided by the COMPLEX query mode and the limitations imposed by the ALL query mode, result ranking only significantly aids the ANY query mode.

The GenSpec module, that ranks records in which a match is found in the genus and species columns higher than the records in which a match is found in other columns does significantly improve results for the ALL and ANY query modes. For the interpreted query mode, results were already better and thus the ranking does not significantly aid performance.

In Table 3, the precision, recall and mean average precision results are found for the unranked birds queries. Due to the unavailability of a freely accessible taxonomic resource for the birds, taxonomic query expansion was not possible. As 30% of the questions mentions a type of material for the birds, such as skin or skeleton, the ANY query mode caused results to explode dramatically as the mention of 'skin' causes already 184,983 results to be returned. Therefore, for the birds only the ALL and COMPLEX search modes are investigated.

As Table 3 shows, contrary to the reptile and amphibian experiments, retrieval is not aided significantly by query expansion. Although indeed recall increases (from 46.83% to 87.22% when using both query interpretation and geographic query expansion), the number of results retrieved

ALL	UnExp	TaxExp	GeoExp	TaxGeoExp
Precision	33.07	**22.84** ▽	**20.92** ▽	32.88 ▽
Recall	31.67	**68.66** ▲	**83.30** ▲	**61.82** ▲
MAP	30.04	**41.45** ▲	**47.61** ▲	**44.78** ▲
ANY	UnExp	TaxExp	GeoExp	TaxGeoExp
Precision	21.62	**15.88** ▽	21.56 ▽	21.62 •
Recall	84.37	84.37 •	84.37 •	84.37 •
MAP	28.28	28.87 ▲	28.87 ▲	28.87 ▲
COMPLEX	UnExp	TaxExp	GeoExp	TaxGeoExp
Precision	40.13	**22.86** ▽	**20.95** ▽	30.38 ▽
Recall	37.59	**69.18** ▲	**85.85** ▲	**54.18** ▲
MAP	35.87	**44.29** ▲	**51.61** ▲	41.14 ▲

Table 1: Precision, recall and mean average precision scores for baseline and expansion modules

ALL	UnExp	TaxExp	GeoExp
GlobalRecordRank	30.27 ▲	**23.81** ▽	**18.25** ▽
LocalRecordRank	30.24 ▲	**27.79** ▽	**19.51** ▽
GenSpec	30.40 ▲	39.77 ▽	41.68 ▽
Unranked	30.04	41.45	47.61
ANY	UnExp	TaxExp	GeoExp
GlobalRecordRank	29.47 ▲	**23.81** ▽	**18.98** ▽
LocalRecordRank	29.17 ▲	23.42 ▽	**19.51** ▽
GenSpec	**42.38** ▲	**39.89** ▲	**39.86** ▲
Unranked	28.28	28.87	28.87
COMPLEX	UnExp	TaxExp	GeoExp
GlobalRecordRank	36.15 ▲	**23.83** ▽	**18.25** ▽
LocalRecordRank	36.11 ▲	**27.80** ▽	**19.49** ▽
GenSpec	36.23 ▲	39.75 ▽	**41.60** ▽
Unranked	35.87	44.29	51.61

Table 2: Mean average precision results expanded ranked reptile and amphibian queries

[4]http://trec.nist.gov/ Last visited: 12 November 2009

	All		Complex	
	UnExp	GeoExp	UnExp	GeoExp
Precision	46.33	**08.66** ▽	46.97	**12.05** ▽
Recall	46.83	**65.42** ▲	47.60	**87.22** ▲
Map	46.17	**12.06** ▽	46.61	**16.36** ▽

Table 3: Precision, recall and mean average precision for unranked expanded bird queries

	ALL		Complex	
	UnExp	GeoExp	UnExp	GeoExp
GlobalRecordRank	46.17 •	13.37 ▽	46.71▲	14.94 ▽
LocalRecordRank	46.17 •	14.15 ▽	46.61 •	20.03 ▲
GenSpec	46.17 •	**19.94** ▲	46.61 •	21.16 ▲
Unranked	46.17	12.06	46.61	16.36

Table 4: Mean average precision results ranked results unexpanded and expanded birds queries

is too great to maintain reasonable mean average precision. This would indicate that the approach does not scale up (as the Birds database is 13 times as large as the Reptiles and Amphibians).

As Table 4 shows, ranking only marginally repairs the mean average precision drop for expanded queries. However, precision, recall and mean average precision do not tell the whole story.

If we look at the results in Tables 5 and 6, we see that, even though the precision drops when query expansion is used, the number of queries for which at least one relevant record is retrieved more than doubles. Thereby, it must also be noted that there are 16 queries for the reptiles and amphibians and birds each, for which there are no relevant records present in the databases. This means that for the reptiles and amphibians only six queries for which a relevant record should have been retrieved remain unanswered, and for the birds only five.

6. DISCUSSION

The performance of Mira is influenced positively by insertion of domain knowledge, in particular recall benefits from it. However, perfect recall is not achieved. There are two reasons for this. First, besides synonyms and different languages, the data also contains spelling errors and non-standard abbreviations. Currently, Mira is not equipped to deal with these. Second, the taxonomic and geographic resources are not complete. For example, the genus *Astylosternidae* is not described in [8], but it is listed in, for example, the Encyclopaedia of Life[5] and the Global Biodiversity Information Facility[6]). Although the Frost taxonomy is an accepted resource there is much ongoing debate on choices made by its author, in particular on the latest version of the taxonomy (personal correspondence with a researcher at Naturalis). However, this is inherent to the nature or systematics and the Frost taxonomy is currently the best to work with. Some geographic terms could also not be matched in GeoNames, for example because GeoNames does not contain many archaic location names, or because the query terms denoted a non-standard spelling.

[5]`http://www.eol.org/` Last visited: July 16, 2009

	UnExp	TaxExp	GeoExp	TaxGeoExp
All	32	66	78	63
Any	78	78	78	78
Complex	38	66	78	54

Table 5: Number of reptiles and amphibians queries for which one or more relevant results are retrieved

	UnExp	GeoExp
All	34	61
Complex	35	81

Table 6: Number of birds queries for which one or more relevant results are retrieved

For the birds, no accepted taxonomic resource was found that could be freely used to expand queries with. The birds experiments were further limited by a scaling problem, namely that Any searches returned too many results and in some cases even almost the whole database. The complex query mode proves itself useful in these experiments as it provides a balance between the strictness of the All query mode and the lenience of the Any query mode. A more open attitude towards data sharing in the natural history domain may help overcome such problems in future work. However, even within these limitations, considerable improvements in retrieval performance are achieved by Mira.

Perfect precision is not obtained because currently the Mira query expansion modules try to expand wherever they can. This may mean that an ambiguous term is expanded by the wrong module. For example, the term 'Anura' will most likely denote the taxonomic name for the order of frogs in the reptiles and amphibians domain. However, in GeoNames, it will also match with *Atoll Kaukura* (which has *Anura* as an alternative name), *Anura Creek* and *Wauna-Anura River*.

Also, some records are returned in which the match with the query term was not found in the most important columns such as the taxonomic columns and the registration number column, but in the longer free-text columns. Excluding the longer free-text columns harms recall, and increases the number of queries that remains unanswered. Therefore the choice was made to include the free-text columns; the GenSpec ranking module was devised to counteract some of the negative influences of this choice, which worked for the Any query mode in particular. Future work will include experiments in which the influence of matches in free-text columns is further restricted.

While the mean average precision scores stay just below 50, in the majority of the queries that we received from the collection managers, the request is for "any specimens of type X". Hence, in many cases, simply finding one relevant record answers the question. When this is taken into account, the fact that for the reptiles and amphibians only 6 queries remain unanswered by using the Mira interpretation and expansion modules, compared to 54 in the baseline system

[6]`http://www.gbif.org/` Last visited: July 16, 2009

and for the birds 3 queries remain unanswered compared to 52 in the baseline system, is a very useful result. Therefore, we can conclude that MIRA provides a significant improvement in access to natural history data.

7. CONCLUSIONS

We have presented a retrieval system that utilises domain knowledge to improve retrieval of natural history data for researchers in this domain. The system utilises domain knowledge in three stages of retrieval, namely (1) query interpretation, (2) query expansion, and (3) result ranking. The result ranking is further aided by the fact that we are querying structured data from a database and not free text. Our results on two databases from the natural history domain show that query interpretation and query expansion improve performance most. Although the system does not attain perfect precision and recall, the knowledge modules diminish the number of queries for which no relevant records are found.

In future work, we plan to fine-tune the expansion modules and diminish the influence of matches in free-text columns by introducing weights in the ranking modules.

8. ACKNOWLEDGMENTS

The authors would like to thank the collection managers at Naturalis for providing data, test queries and expert knowledge about the domain and the anonymous reviewers for their comments. The research reported in this paper was funded by the Netherlands Organization for Scientific Research (NWO) in the project Mining for Information in Texts from the Cultural Heritage (MITCH), grant number 640.002.403.

9. REFERENCES

[1] E. Agirre and A. Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics (EACL-2009)*, 2009.

[2] A. Balmin, V. Hristidis, and Y. Papakonstantinou. ObjectRank:authority-based keyword search in databases. In M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, editors, *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB 2004)*, Toronto, Canada, August 31 - September 3 2004. Morgan Kaufmann.

[3] R. C. Bodner and F. Song. Knowledge-based approaches to query expansion in information retrieval. In *Proceedings of the 11th Biennial Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, pages 146–158. Springer-Verlag London, UK, 1996.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engines. *Computer Networks and ISDN Systems*, 30:1–7, 1998.

[5] S. Büttcher, C. L. A. Clarke, and G. V. Cormack. Domain-specific synonym expansion and validation for biomedical information extraction (multitext experiments for trec 2004). In *Proceedings of the 13th Text Retrieval Conference*, 2004.

[6] S. Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. In *Proceedings of the 16th international conference on World Wide Web*, pages 571–580, Banff, Alberta, Canada, 8–12 May 2007. ACM New York, NY, USA.

[7] R. Diestel. *Graph Theory*. Graduate texts in matematics. Springer-Verlag, Berlin, New York, 3rd edition, 2005. ISBN 978-3-540-26183-4.

[8] D. R. Frost. Amphibian species of the world: an online reference. version 5.3. Electronic Database accessible at http://research.amnh.org/herpetology/amphibia/, February 2009. American Museum of Natural History, New York, USA.

[9] N. Guarino, C. Masolo, and G. Vetere. OntoSeek: Content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80, May/June 1999.

[10] T. H. Haveliwala. Topic-Sensitive PageRank. In *Proceedings of WWW2002*, Honolulu, Hawaii, USA, May 7-11 2002. ACM Press New York.

[11] W. Hersch and E. Voorhees. TREC genomics special issue overview. *Information Retrieval*, 12(1):1–15, Feb 2009. Special Issue on TREC Genomics.

[12] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[13] P. Lendvai and S. Hunt. From field notes towards a knowledge base. In *Proceedings of LREC 2008*, Marrakech, Morocco, 2008.

[14] J. Lin and D. Demner-Fushman. The role of knowledge in conceptual retrieval: A study in the domain of clinical medicine. In *Proceedings of SIGIR'06*, pages 99–106, Seattle, Washington, USA, August 6-11 2006. ACM, ACM New York, NY, USA.

[15] D. Milne, I. H. Witten, and D. M. Nichols. A knowledge-based search engine powered by wikipedia. In *Proceedings of CIKM'07*, pages 445–454, Lisbon, Portugal, November 6-8 2007. ACM, ACM New York, NY, USA.

[16] R. Navigli and P. Velardi. An analysis of ontology-based query expansion strategies. In *Proceedings of 2003 Workshop on Adaptive Text Extraction and Mining (ATEM'03)*, pages 42–49, Cavtat-Dubrovnik, Croatia, September 22- 26 2003.

[17] K. Spärck-Jones. *Automatic Keyword Classification for Information Retrieval*. Buttersworth, 1971.

[18] S. Tata and G. M. Lohman. SQAK: doing more with keywords. In *Proceedings of SIGMOD 2008*, 2008.

[19] T. Tran, P. Cimiano, S. Rudolph, and R. Studer. Ontology-based interpretation of keywords for semantic search. In *Proceedings of ISWC 2007*, 2007.

[20] P. Uetz, J. Goll, and J. Hallermann. The reptile database. http://www.reptile-database.org, 2008. Last visited: June 4, 2009.

[21] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69, Dublin, Ireland, 1994. USA.

[22] Q. Zhou, C. Wang, M. Xiong, H. Wang, and Y. Yu. SPARK: Adapting keyword query to semantic search. In *Proceedings of ISWC 2007*, 2007.

Semantic vs term-based query modification analysis

Vera Hollink
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
V.Hollink@cwi.nl

Theodora Tsikrika
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
Theodora.Tsikrika@cwi.nl

Arjen de Vries
Centrum Wiskunde en Informatica
Science Park 123
1098 XG Amsterdam
Arjen.de.Vries@cwi.nl

ABSTRACT

Previous research has studied query modifications on a syntactic level by focusing on the addition, elimination and substitution of terms between consecutive queries that have at least one term in common. In this paper, we determine semantic relations between queries by first mapping them onto concepts in linked data sources and then identifying the relations between the concepts. This enables us to find relations between queries that do not share any terms. Moreover, with this approach we can find more detailed and more meaningful query modification patterns than with a term-based analysis. Application of our method to search logs of two search engines shows the importance of studying query modifications on a semantic level. Our results indicate that users often search for entities that are related semantically, but not syntactically. Specifically, users often successively search for two entities sharing a common property, such as two actors starring in the same movie, or two entities with a specific relation, such as spouses. We discuss the implications of these findings for the design of search engines.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Query formulation*

Keywords

Query modification, linked data, query log analysis

1. INTRODUCTION

Users of search engines often engage in an iterative interactive process by providing a succession of queries so as to satisfy a single information need. This search process is typically structured as follows: the user formulates and submits a query, examines the retrieval results, and then, depending on his (or her) satisfaction with the results, decides to either stop or to enter a new search cycle by modifying the query and re-submitting it in an attempt to reach a better outcome. Given that query modification is a key user behavior [10, 3, 14, 11], retrieval systems should provide assistance to their users during this challenging task. The study of query modification patterns allows us to gain insights into this user behavior, which can be used to improve the support that search engines offer to their users.

Query modification patterns are usually identified through the analysis of queries collected in search interaction logs [9]. Previous studies (e.g., [10, 14, 11]) have classified query modifications based on the overlap between terms in consecutive queries by examining whether terms have been added, eliminated, or substituted in a query, and interpreting additions as specifications, eliminations as generalizations, and substitutions as reformulations. The major limitation of such term-based methods is that they can only classify pairs of queries that have at least one term in common and, therefore, cannot determine the relations between queries that are semantically related without sharing any terms, such as the queries `Wim Kok` and `Ruud Lubbers`. Furthermore, such methods do not typically make any finer distinctions within each of the three main classes, even though there exist subclasses that correspond to very different user intents. For example, the modification of query `Posthuma Tour France` to `Posthuma Tour 2008` most likely indicates a second attempt to find information about the same event, while the modification of `Candy Dulfer` to `Hans Dulfer` signifies a shift of attention to another person. Nevertheless, term-based methods classify both cases as reformulations.

The work presented in this paper is also concerned with the identification of query modification patterns through the analysis of search logs. However, it goes beyond the term-based methods and proposes an approach that determines semantic relations between queries by exploiting the knowledge in a *linked data cloud* [1, 2]. Combining the semantic relations with statistics from the search logs allows us to recognize fine-grained and meaningful query modification patterns that are not visible from the usage statistics alone. For instance, the use of DBpedia[1] allows us to detect that many users successively search for two entities that have some property in common, such as both being soccer players in the same national team. In this paper we present our method for detecting semantic modification patterns and discuss the implication of such patterns for the design of search engines.

The remainder of this paper is structured as follows. In

[1] http://dbpedia.org/

Section 2 we present related work on query modification analysis and briefly explain the key ideas behind linked data. Section 3 presents our approach. In Section 4 we present results of applying our approach to the search logs of two search engines. In addition, we compare our findings to the results of a term-based analysis. The last section contains conclusions and discusses our results.

2. RELATED WORK

A number of studies have classified query modification patterns encountered in search logs of various types of search engines. Probably the most studied search logs are those of the Excite search engine [14, 15, 13]. Other general purpose engines that have been analyzed include Dogpile [10] and Yahoo! UK [3] and Yahoo! US [3]. Other researchers have examined the logs of search engines for limited domains, such as intranets [7] or commercial image providers [11]. Finally, Huang et al. [8] analyze logs of a group of users accessing a variety of search engines via a proxy.

Query modifications can be classified either manually [5, 13, 11, 14] or automatically [7, 15, 10, 3]. Manual classification is necessarily limited to a small number of queries, ranging from 2109 queries in [14] to 4690 queries in [13]. Automatic methods enable the analysis of much larger samples, up to 16 million queries in [3].

Studies that employ automatic methods usually classify query modifications solely on the basis of terms in the queries. These studies (as well as some of the manual studies) examine whether queries have been added, eliminated or substituted compared to the user's previous query [10, 15, 11, 7, 5]. When terms are added, the search is considered to become more specific (e.g., from query `Beatrix` to query `Beatrix holiday`), when terms are eliminated the search becomes more general (from `Beatrix holiday` to `Beatrix`) and when terms are substituted a parallel movement is made (reformulation) (from `Beatrix 2008` to `Beatrix 2009`). Some of the manual studies do not only look at terms but also classify modifications based on the meaning of the queries [14, 11, 13]. In these studies the same main classes are used, but a semantic modification from `Dog` to `Labrador` is also classified as a specification. An interesting intermediate approach is presented in [3]: this work aims at a semantic classification of query modifications into specification, generalization and reformulation by looking at the overlap in query terms, time intervals between queries and features of the user session as a whole.

The large majority of the studies find that the most frequently used modification type is reformulation, followed by specification and generalization [5, 13, 11, 14, 15, 10, 3]. Reformulations occur roughly twice as often as specifications which occur twice as often as generalizations. This finding is also supported by the work in [8], where it is observed that queries in the beginning of sessions tend to be more general than later queries. A noteworthy observation is that there is no difference in this respect between manual and automatic methods or between purely term-based and semantic methods. The only study in which different proportions are found is [7]: they find an equal number of reformulations and specifications and a much smaller number of generalizations. No explanation is given for this deviation.

Two studies have looked at the number of times users enter term variations [14, 5]. Such variations include, for instance, modifications from singular to plural forms or vice versa. Term variations are less common than the main modification classes, but they still make up a significant proportion of the queries, occurring about half as frequently as generalizations.

Sequences of query modifications are examined in [15, 10, 3]. Whittle et al. [15] found that users tend to repeat the same modification type (e.g., a specification is often followed by another specification). This is not confirmed by [3], who found that specifications are usually followed by generalizations and generalizations by specifications. This is also the dominant pattern in [10].

Lau and Horvitz [13] examined the relation between modification types and the time interval between submitting two consecutive queries. They find that specification is most likely after an interval of 20 to 30 seconds while reformulation peaks when the interval is longer than 5 minutes. However, the differences are quite small.

In summary, the four term-based modification patterns (specification, generalization, reformulation and term variation) have been extensively studied. Different authors have researched different variants and aspects of these patterns, but to our knowledge there are no papers that classify query modifications on a semantic level.

The key element that sets our approach apart from existing approaches and enables it to find semantic query modification classes is the use of linked-data. Below we will briefly review the main concepts of linked data. For a extensive overview we refer to [2].

The idea of linked data was first described by Tim Berners-Lee [1] in the form of four principles that prescribe how data should be published on the web. Following these principles ensures that the data can be easily shared with others, read by both humans and machines and linked to data from other sources. Each entity in the data is referred to by a unique URI. Information about the entity can be attained by looking up the URI via HTTP. Information about entities and links between entities are coded in RDF [12]: a set of triples <subject, predicate, object>, where the subject and the predicate are both URIs and the object can either be a URI or a string literal. Examples of RDF triples are given in Figure 1. The first triple provides information about a single entity and says that the concept `synset-soccer_player-noun-1` is described by the label `'soccer player'`. The second triple provides a link between entities from different sources, stating that Edwin van der Sar has the type soccer player in WordNet[2]. The third triple states that two URIs from different sources refer to the same entity.

The Linked Open Data Project[3] aims to publish and connect as many open data sets as possible according to the linked data principles. Since the start of the project the number of data sets has grown explosively. The current size of the

[2]http://www.w3.org/2006/03/wn/wn20/

[3]http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

```
Subject:    http://www.w3.org/2006/03/wn/wn20/instances/synset-soccer_player-noun-1
Predicate:  http://www.w3.org/2006/03/wn/wn20/schema/senseLabel
Object:     'soccer player'

Subject:    http://dbpedia.org/resource/Edwin_van_der_Sar
Predicate:  http://dbpedia.org/property/wordnet_type
Object:     http://www.w3.org/2006/03/wn/wn20/instances/synset-soccer_player-noun-1

Subject:    http://e-culture.multimedian.nl/ns/rijksmuseum/people5706
Predicate:  http://www.w3.org/2004/02/skos/core#exactMatch
Object:     http://e-culture.multimedian.nl/ns/getty/ulan#500011051
```

Figure 1: Examples of RDF triples

total data cloud is estimated at 4.7 billion triples [2]. The two largest data sets that we use in the experiments in this paper, DBpedia[1] and WordNet[2], are taken from this cloud.

3. METHOD

To determine how users of a search engine modify their queries, we extract the queries of individual users from a search log file. We map the queries on concepts in a linked data cloud and search the linked data to determine the semantic relation between pairs of consecutive queries submitted by the same user. Finally, we count how often each type of relation occurs. In the rest of this section, these steps are explained in more detail.

3.1 Preprocessing

Before the actual analysis can take place the server logs of the search engine that is analyzed must be preprocessed. Cooley et al. [6] have described the various preprocessing steps in depth. Here we give a brief summary of the elements that are relevant for our purposes.

Server logs of search engines typically contain an entry for each query that is submitted through the engine and for each click that a user has made on a search result. Among other things, a log entry consists of the user's IP address, information about the browser that was used (called the *agent*), the time of the request and the submitted query or clicked result. Sometimes additional information about users is available, coming, for instance, from browser cookies or log-in mechanisms.

We group the log entries per user. If cookies or log-ins are available users can be identified with certainty. Otherwise, we assume that all entries with the same IP address and agent are from one user. A new user session starts when there is a period of inactivity in the session longer than some predefined time interval (typically 15 or 30 minutes). Finally, we list for each session the queries that are entered and conflate consecutive identical queries into one query.

3.2 Finding semantic relations between queries

The queries in the user sessions are mapped on concepts in a linked data cloud. Finding relevant concepts for queries is far from trivial, as it is often not clear what a user is exactly looking for when entering a query. In this study we use the `rdfs:label` property of the concepts in the linked data to match the queries, as this property is meant to provide a human readable description for the concepts [4]. We map queries on concepts that have an `rdfs:label` that exactly matches the query. If no exact match can be found, queries are mapped onto concepts with labels that contain all query terms (after stemming). With this method each query is mapped onto zero, one, or multiple concepts. We purposely chose a conservative mapping method, sacrificing recall for precision, to reduce the amount of noise in the resulting modification patterns.

For each pair of queries that are consecutively submitted by the same user, we determine the semantic relation between the queries, as illustrated in Figure 2. A graph search algorithm is used for traversing the links in the linked data to find the shortest series of links that connects the two queries (their relation). As linked data graphs are often very large, measures have to be taken to keep the search tractable. We set the maximum number of links in a relation at 4. Pilot experiments showed that longer relations are hardly ever relevant. Furthermore, we remove all concept-predicate pairs from the linked data that were present in more than 10,000 triples, as these relations are usually overly generic (e.g., stating that a person's gender is male).

Equivalence relations, such as `skos:exactMatch` and `owl:sameAs`, indicate that two URIs refer to the same entity. The path search algorithm treats such equivalent entities as one (they are *smushed*). In other words, the equivalence relations are not reported in the relation between the queries and are not counted in the number of links that is followed.

Often multiple relations of the same length are found between two queries. For instance, two persons can both be of type soccer player and also both play in the same national team. All relations that are found for a pair of queries are taken into account, but in the rest of the analysis each relation receives a weight that is inversely proportional to the number of relations that are found for the query pair.

3.3 From relations to modification patterns

The next step is to abstract away from relations between specific instances and infer *modification patterns* by removing the instances and keeping just the links. For instance, we may find that the relation from query `David Beckham` to query `Joe Cole` is that both refer to players in the English national football team:

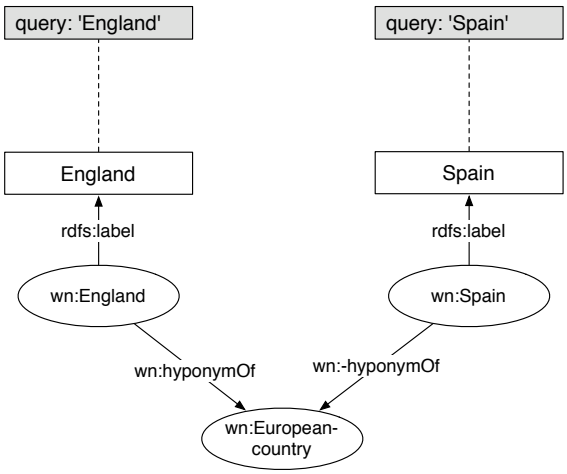


Figure 2: Example application of our procedure: the relation between queries `England` and `Spain` is that they both match hyponyms of the WordNet concept `European country`.

```
David Beckham –DBpedia:nationalteam→
England_national_football_team
←DBpedia:nationalteam– Joe Cole
```

The arrows denote the directions of the predicates. This relation is abstracted to the modification pattern:

```
Q1 –DBpedia:nationalteam→ X
←DBpedia:nationalteam– Q2
```

To determine the importance of the patterns that are found, we count how often each pattern occurs between queries in the search log. In this count all sessions are weighted equally, so that sessions with more queries do not contribute more to the final counts than shorter sessions. The *support* of a pattern is defined as its relative frequency.

The support value of each pattern is compared to a baseline that represents the expected frequency of the pattern. The baselines are computed by randomly sampling pairs of queries from different sessions in the log file and determining the relations between these pairs. We define the *confidence* of a pattern as the proportion of all (inter- and intra-session) query pairs matching the pattern that come from the same search session. Thus, if a pattern occurs in 3% of the query pairs where both queries come from the same session and 0.6% of the query pairs where the two queries come from different sessions, its support is 0.03 and its confidence is 0.03/(0.03+0.006)=0.83.

Finally, we apply an iterative process to improve the accuracy of the relations that we found. Patterns with high support but low confidence occur equally often between pairs from the same session as between pairs from different sessions, and thus are with high probability irrelevant patterns. We look up all query pairs for which relations are found that match an irrelevant pattern. We discard these relations and search the graph for other (longer) relations between the queries. When the new relations are determined, we recompute support and confidence. This process continues until the support and confidence of all patterns are above given thresholds or until no more relations are found. Finally, we output all patterns that are likely to be highly relevant, i.e. the patterns having both high support and confidence.

4. CASE STUDIES

4.1 Data sets

We applied our semantic query modification method to the search log files of the commercial picture portal of a European news agency. The portal provides access to more than 2 million photographic images covering a broad domain. The log files record the search interactions of professional users (mainly journalists) accessing the picture portal. We used one year of search logs, containing 1,105,766 queries in 332,809 sessions. Search sessions were identified using a log-in and a browser cookie. The linked data consisted of various interlinked sources: the DBpedia Ontology[1], WordNet[2], the Cornetto Lexical Knowledge Base[4] (which contains both Dutch and English terms), the Getty[5] Thesaurus of Geographical Names, and the Getty Art and Architecture Thesaurus (aat). Together these collections comprise 22 million RDF triples.

The second search engine is the search facility of the Rijksmuseum web site[6] (Rijks), a Dutch art museum. The log files cover 5.5 months and consist of 216,217 queries in 45,046 sessions, where sessions were identified using IP addresses and agent fields. As linked-data, we used WordNet, Cornetto, the Dutch version of the Getty thesauri, and also various Dutch art-specific ontologies that were collected and interlinked in the E-Culture project[7].

For both data sets the support threshold was set at 0.0005 and the confidence threshold at 0.66667.

4.2 Semantic modification patterns

For 55% of the 482,717 query modifications in the News photo data set and 46% of the 49,410 query modifications in the Rijksmuseum data set, the two queries could both be mapped onto concepts in the linked data (see Table 1). In both data sets a relation was found for about half of the modifications for which concepts were found. On average 12.5 (News photo) and 6.2 (Rijksmuseum) relations were found per query pair.

	News photo	Rijksmuseum
no concept found	0.45	0.54
no relation found	0.30	0.23
relation found	0.25	0.23

Table 1: Proportion of query pairs for which a relation could (not) be found in the linked data.

We manually evaluated the concepts that were found for 100 random queries from the News photo data set. For 74% of

[4]http://www2.let.vu.nl/oz/cltl/cornetto/
[5]http://www.getty.edu
[6]http://www.rijksmuseum.nl/
[7]http://e-culture.multimedian.nl/

Table 2: The 10 semantic modification patterns with the highest support in the News photo data set.

	support	confidence	pattern
1.	0.031	0.94	`[ ]`
2.	0.017	0.99	`Q1 −DBpedia:spouse→ Q2`
3.	0.017	0.99	`Q1 ←aat:distinguished_from− Q2`
4.	0.017	0.86	`Q1 −DBpedia:birthplace→ X ←DBpedia:birthplace− Q2`
5.	0.013	0.91	`Q1 −rdf:type→ X ←rdf:type− Q2`
6.	0.012	0.95	`Q1 −DBpedia:nationalteam→ X ←DBpedia:nationalteam− Q2`
7.	0.009	0.99	`Q1 −DBpedia:partner→ Q2`
8.	0.009	0.90	`Q1 −DBpedia:wordnet_type→ X ←DBpedia:wordnet_type− Q2`
9.	0.008	0.96	`Q1 −aat:distinguished_from→ Q2`
10.	0.008	0.96	`Q1 −WordNet:memberMeronymOf→ X ←WordNet:memberMeronymOf− Q2`

Table 3: The 10 semantic modification patterns with the highest support in the Rijksmuseum data set.

	support	confidence	pattern
1.	0.123	0.96	`[ ]`
2.	0.020	0.74	`Q1 −WordNet:hyponymOf→ X ←WordNet:hyponymOf− Q2`
3.	0.008	0.73	`Q1 −Cornetto:domain→ X ←Cornetto:domain− Q2`
4.	0.003	0.92	`Q1 −rdf:type→ X ←rdf:type− Y ←Rijks:material− Z −Rijks:schilder→ Q2`
5.	0.003	0.87	`Q1 −Cornetto:hasHyperonym→ X ←Cornetto:hasHyperonym− Q2`
6.	0.003	1.00	`Q1 ←Cornetto:hasHyperonym− Q2`
7.	0.003	1.00	`Q1 −WordNet:hyponymOf→ Q2`
8.	0.002	1.00	`Q1 −Cornetto:hasHyperonym→ Q2`
9.	0.002	0.72	`Q1 −Getty:nationalityNonPreferred→ X ←Getty:nationalityNonPreferred− Q2`
10.	0.002	0.71	`Q1 −Cornetto:domain→ X ←Cornetto:domain− Y −Cornetto:eqNearSynonym→ Q2`

the queries a matching concept was present in our linked data. Our mapping method found a concept for 72% of the queries. For 89% of these queries at least one correct concept was found (precision). For some queries multiple concepts were found. When a correct concept was found, on average 85% of the concepts found were correct. Recall was 86%, meaning that in 86% of the cases in which a correct concept was in the linked data, it was also found. These results suggest that our mapping method is quite accurate, despite its simplicity.

We also evaluated for 100 random query pairs the relations that were found. Our method found a relation for 39% of the query pairs and for 51% of these pairs at least one correct relation was found (precision). For pairs where a correct relation was found, the majority of the relations that were found were correct (74%). Recall was 63%. Although finding relations proved more difficult than finding concepts, we believe our method is accurate enough to find reliable query modification patterns. The incorrect relations appear to be more or less random, so that the patterns that occur in high numbers are in majority based on correct paths.

The support and confidence thresholds effectively removed many irrelevant patterns. An example of a discarded pattern in the News photo data is:

```
Q1 −rdf:type→ X ←rdf:type− Y −DBpedia:
birthplace→ Z ←DBpedia:birthplace− Q2
```

This pattern applies to query pairs consisting of two persons `Q1` and `Q2`, so that there exist some person `Y` who is of the same type as `Q1` (e.g., both tennis players) and is born in the same place at `Q2`. This pattern occurred quite frequently between queries from the same session (support 0.0015), but even more frequently between queries from the different sessions (confidence 0.39) and thus was discarded.

The patterns with the highest support in the News photo data are shown in Table 2. The most common pattern was the identity relation ([ ]): two different queries that are mapped to the same concept, usually variant names for the same entity, such as 'Gent' and 'Gand' (the Dutch and French name of a Belgian City). Patterns 2 and 7 indicate that many users searched first on the name of a person and then on the name of his or her spouse or partner. Pattern 6 tells us that many users searched on two people from the same national team. Patterns 5 and 8 both say that users searched on two entities from the same type, such as tennis players or townships. Pattern 10 shows that people often search for two entities that are part of the same whole. Inspection of the query pairs that follow this pattern shows that these are mainly entities that are both part of the concept `WordNet:royalty` (e.g., queries `princess` and `king`).

Compared to the News photo data, the Rijksmuseum data shows more patterns that involve concepts named by common nouns (e.g., patterns 2, 5, 6, 7 and 8 in Table 3) and less concepts named by proper nouns (e.g., pattern 9 in Table 3). This difference has consequences for the support that should be offered by the search engines. Users of the Rijksmuseum web site would probably be helped best by showing terms that are related to their current query via relations from a thesaurus. The News photo users, on the other hand, would most likely benefit more from showing entities that are related to their current query via domain specific relations.

Another difference between the Rijksmuseum patterns and the News photo patterns, is that in the Rijksmuseum data

the identity relation ([ ]) has a much higher support. This shows that users of this search engine have more trouble formulating their query and thus try more variant names before finding the right name for the entity they search for. The search engine could support this by offering a list of spelling variants whenever the current query does not yield any results.

In both data sets a large proportion of the relations were *sibling relations*: relations of the form `Q1–R→X←R–Q2`. Examples include patterns 4, 5, 6, 8 and 10 in Table 2 and patterns 2, 3, 5 and 9 in Table 3. In the News photo data set these patterns made up 22% of the determined relations; in the Rijksmuseum data set 9%. This shows that many people search for two entities with some common property, such as two actors starring in the same movie or two hyponyms of a WordNet concept. This finding is in line with the work of Rieh and Xie [14], who found through manual classification of query modifications that sibling relations (referred to as 'parallel movements' by them) were the most common modification type. However, their analysis did not identify what kind of siblings were used (actors, or soccer players, or hyponyms, etc.).

Another frequently occurring relation type (14% News photo, 2% Rijksmuseum) are *direct few-to-few relations*, such as 'spouse-of' and 'has-capital'. Here few-to-few relations are defined as a relaxed version of one-to-one relations, where 'few' means on average less than 2.

Finally, we observe that most of the relations that are found (74% News photo, 68% Rijksmuseum) consist of more than one link. Apparently, users often search for entities that are related in a complex way.

4.3 Comparison with term-based analysis

For comparison, we also analyze the data sets with a term-based approach (see Section 2). First, the query terms are stemmed. Then, for each stemmed query we determine whether compared to the previous query terms are added (specification), removed (generalization) or replaced (reformulation). In addition, we count how many times stemming made the query the identical to the previous query (remember from Section 3.1 that consecutive queries that were identical before stemming are removed). Query pairs without overlapping terms are classified as 'undetermined'. The frequency of each type of term modification is shown in Table 4 and Figure 3.

In the News photo data reformulations occur most frequently, followed by specifications, generalizations and stem-identical queries. These findings closely match the findings of previous studies [5, 13, 11, 14, 15, 10, 3]. In the Rijksmuseum data we find the same pattern except that in these data there are a relatively large number of specifications. This distribution is in line with the findings in [7], but it is unclear why it differs from the other data sets.

modification type	News photo	Rijksmuseum
specification	0.08	0.13
generalization	0.05	0.07
reformulation	0.11	0.10
stem-identical	0.01	0.03
undetermined	0.75	0.67

Table 4: Frequency of term-based query modification patterns.

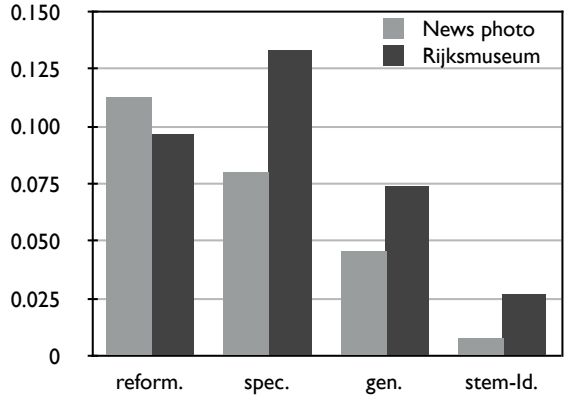


Figure 3: Relative frequency of term-based modification types.

25% of the modifications in the News photo data and 33% of the modifications in the Rijksmuseum data can be assigned to one of the four term-based classes. These percentages are comparable to the percentages of cases that could be assigned to a semantic modification class (see Table 1). However, as shown in Figure 4 the two approaches classify different query pairs: the linked data approach found a relation for only 9% (News photo) and 19% (Rijksmuseum) of the cases that were classified by the term-based approach. Conversely, the term-based approach found a class for only 9% (News photo) and 27% (Rijksmuseum) of the cases for which a semantic relation was found. One reason for this effect is that the term-based approach works well for query pairs consisting of multiple entities, such as `Beatrix`, and `Beatrix holiday`, but cannot handle most pairs consisting of single entities, such as `Beatrix`, and `Willem-Alexander`. The linked-data approach, on the other hand, can handle single entity queries, but not multiple entity queries. This indicates that the two approaches are to a large extent complementary.

Reformulations are related to sibling relations: for example, the modification from the query `elm tree` to the query `oak tree` is both a reformulation and a sibling relation. However, a large part of the siblings cannot be recognized by looking at reformulations: only 6% (News photo) and 2% (Rijksmuseum) of the siblings were recognized as reformulation. Inspection of the results shows that many siblings consist of names of two persons, such as two players in the same national team. The names do not have any terms in common and thus are not classified as reformulations. In other words, the queries were semantically related but not in terms of terms.

There is no corresponding term-based class for direct few-to-few relations. These relations are classified as reformulations, specifications, generalizations and stem-identical, but most often as undetermined.

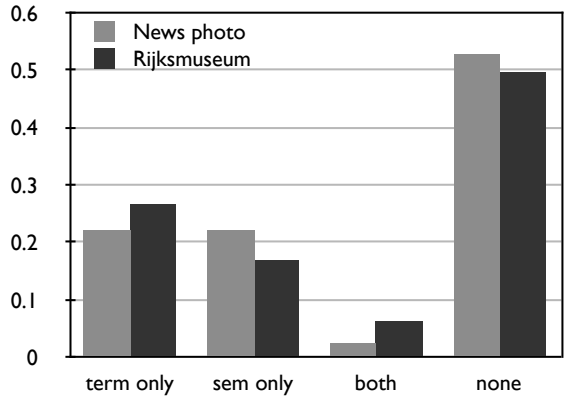


Figure 4: Overlap between the query pairs classified by the semantic and the term-based approach: the proportion of queries for which a relation is found only by the term-based approach, only by the semantic approach, by both approaches, and by none of the approaches

In conclusion, there are types of modifications that appear to be important for users, but that cannot be identified with a term-based analysis of query modifications. Conversely some modifications that can be classified with a term-based approach cannot currently be classified using linked data, most notably modifications involving queries with multiple entities. Thus, the linked data approach does not make a term-based analysis obsolete, but forms a valuable addition to it.

5. CONCLUSIONS

In this paper we showed the potential of combining statistical information gathered from log files with semantic information from linked data. The use of linked data for query log analysis enabled us to find patterns in query modification behavior that are interesting, non-trivial and easy to interpret. In contrast to traditional term-based approaches, the linked data approach finds relations between queries that do not have any terms in common. This is a large advantage as our analysis indicates that users often search for two entities sharing a common property (e.g., both being tennis players) or two entities with a direct relations (e.g., spouses). These entities are related semantically, but do not have common terms. Moreover, with the linked data approach we can find more detailed modification patterns than with term-based approaches.

Insights gained from semantic query modification analysis can be used directly for improvement of search engines. We found that users often try various names to find the same entity indicating that users are often unsure about which names are used in the data set. Search engines can support this by showing a list of variants of the current query that occur in the data set. Furthermore, we found that users of one search engine mainly modified their queries according to domain-specific relations, such as partner-of, while users of another search engine tended to use thesaurus relations such as hyponym-of. Knowing which types of relations are important for the users of a search engine, enables us to offer search support that is tailored to the user population of that search engine.

One form of search support are suggestions for follow-up queries. In the next step of our research we will apply semantic query modification patterns to query suggestion. If we know that users who search on the name of a soccer player often also want information about other players from the same team, we can suggest the names of other players to a user who searches on the name of one player. In contrast to purely statistical query suggestion methods, these types of patterns can also be applied to queries that are entered for the first-time.

Another direction of further research will be the extension of our analysis from query pairs to query sequences. By examining search sessions as a whole, we can reveal session-wide search patterns that extend beyond individual query pairs.

6. REFERENCES

[1] T. Berners-Lee. Linked data: Design issues. http://www.w3.org/DesignIssues/LinkedData.html, 2006. last accessed November 5, 2009.

[2] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems, Special Issue on Linked Data*, in press.

[3] P. Boldi, F. Bonchi, C. Castillo, and S. Vigna. From 'dango' to 'japanese cakes': Query reformulation models and patterns. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Milan, Italy*, pages 183–190, 2009.

[4] D. Brickley and R. V. Guha. RDF vocabulary description language 1.0: RDF schema. http://www.w3.org/TR/rdf-schema/, 2004. last accessed November 5, 2009.

[5] P. Bruza and S. Dennis. Query reformulation on the internet: Empirical data and the hyperindex search engine. In *Proceedings of the RIAO'97 Conference on Computer-Assisted Searching on the Internet, Montreal, Canada*, pages 488–499, 1997.

[6] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1:5–32, 1999.

[7] D. He, A. Göker, and D. J. Harper. Combining evidence for automatic web session identification. *Information Processing and Management*, 38(5):727–742, 2002.

[8] C.-K. Huang, L.-F. Chien, and Y.-J. Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology*, 54(7):638–649, 2003.

[9] B. J. Jansen. Search log analysis: What it is, what's been done, how to do it. *Library and Information Science Research*, 28(3):407–432, 2006.

[10] B. J. Jansen, D. L. Booth, and A. Spink. Patterns of query reformulation during web searching. *Journal of the Americal Society for Information Science and Technology*, 60(7):1358–1371, 2009.

[11] C. Jörgensen and P. Jörgensen. Image querying by image professionals. *Journal of the Americal Society for Information Science and Technology*, 56(12):1346–1359, 2005.

[12] G. Klyne and J. J. Carroll. Resource description framework (RDF): Concepts and abstract syntax. http://www.w3.org/TR/rdf-concepts/, 2004. last accessed November 5, 2009.

[13] T. Lau and E. Horvitz. Patterns of search: analyzing and modeling web query refinement. In *Proceedings of the Seventh International Conference on User Modeling, Banff, Canada*, pages 119–128, 1999.

[14] S. Y. Rieh and H. Xie. Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Information Processing and Management*, 42(3):751–768, 2006.

[15] M. Whittle, B. Eaglestone, N. Ford, V. J. Gillet, and A. Madden. Data mining of search engine logs. *Journal of the American Society for Information Science and Technology*, 58(14):2382–2400, 2007.

The Impact of Summaries: What Makes a User Click?

Khairun Nisa Fachry[1] Jaap Kamps[1,2] Junte Zhang[1]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] ISLA, Faculty of Science, University of Amsterdam

ABSTRACT

Modern retrieval systems are in fact two-tier systems in which a user first views summaries of the results in a hit-list, and only when she decides to "click," the full result document is consulted. Standard information retrieval evaluation ignores the crucial summary step, and directly evaluates in terms of the relevance of the resulting document. In this paper, we investigate the impact of the result summaries on the user's decision to click or not to click. Specifically, we want to find out both what information in the summary triggers a positive selection decision to view a result, and what information triggers a negative selection decision. We use a special document genre, archival finding aids, where results have a complex document structure and currently available systems experiment with structured summaries having both static elements (like the title and a manually compiled abstract by an archivist) and query-biased snippets (showing the matching keywords in context). We conducted an experiment in which we asked test persons to explicitly mark the parts of summaries that trigger a selection decision, and asked them to explain further (i.e. why and how). The results from this user study indicate the importance of sufficient context in the summary. Selection decisions were primarily based on the static elements: the title and abstract of the document. This may be a result of the completeness and coherence of the information in these elements, although also the length played a clear role. A whole paragraph (as in the abstract) triggered a decision more frequently than a short sentence (as in the title) or an incomplete sentence (as in the query-biased snippets).

1. INTRODUCTION

Modern information retrieval systems are in fact two-tier systems. Imagine a scenario about a user with a particular information need. In the first stage, she will inspect summaries of the results in a hit-list and tries to assess which results potentially satisfy her information need. Based on a promising summary, she may decide to "click" and enter a second stage in which she consults the full result document looking for useful information given her information seeking need. In these two-tier systems, the summaries on the hit-list play a crucial role and act as a filter: only when the summary is deemed adequate, the result is inspected.

Standard information retrieval evaluation ignores this crucial summary step and directly evaluates in terms of the relevance of the resulting document. Turpin et al. [20], in their study of including summaries in system evaluation, revealed that summaries need to be evaluated in addition to the document when constructing a test collection. In their experiment, in which users were asked to provide relevance assessments of both summaries and documents, 14% of the highly relevant and 31% of relevant documents were never examined by the users because the summary was judged irrelevant. This shows that the document summary presented by a retrieval system does not always accurately reflect the document content. Since summaries evaluation is the first selection moment for the users, this could results in users missing out some relevant documents.

In this paper, our main aim is to investigate the impact of the summaries of documents on a user's decision to either click or not. Specifically, we investigate the following two research questions:

1. What information in the summaries triggers a positive selection decision to view a result and what information triggers a negative selection decision?
2. Why and how does this influence the decision to click or not to click?

We research these questions for a special document genre, archival finding aids. Archival finding aids are descriptions of archival collections. Since archival collections can be huge, their descriptions may cover 100s of pages. Archival descriptions are structured in a hierarchical way, from general (an overview description of the whole collection) to the specific (a description at the lowest level, most commonly file or item level). Archival finding aids are increasingly encoded in an Extended Markup Language [XML, 21] format called Encoded Archival Description [EAD, 7], which is the *de facto* standard. Archival descriptions are an interesting special case for result summarization, since the documents themselves are long in content and complexly structured. In particular, the descriptions contain various fields such as the title and a human-generated abstract (summary of the whole collection). In addition, short teasers or snippets showing keywords in context can be derived from the textual content of the document. Tombros and Sanderson [19] demonstrated that these can significantly improve both the accuracy and speed of user relevance judgments. Hence, the archival descriptions allow us to experiment with both static elements and query-biased snippets in the result summaries.

We conducted an experiment in which we asked test persons to explicitly mark the parts of summaries that trigger

DIR-2010 January 25, 2010, Nijmegen, the Netherlands.

selection decisions, and asked them to explain why this information triggered their decision. To answer our first research question, we look at two outcomes of a selection decision, i.e. a positive and negative selection decision. For each decision, we count the part of the summaries marked by the test persons and this results in quantitative data. To answer our second research question, we look at the qualitative data on why and how the information triggered the decision.

The remainder of the paper is organized as follows. In Section 2, we describe related work on selection decisions. In Section 3, we describe the methodology of the user study. In Section 4, we describe the result of the user study. Finally, in Section 5 we discuss the results and draw our conclusions.

2. RELATED WORK

In this section, we will discuss related work on selection decisions in literature search, in XML retrieval, and in archival access.

2.1 Selection Decision in Literature Search

A selection decision is based on the (assumed) relevance of a result. The concept of relevance is fundamental in information retrieval, and has attracted continual interest. See Saracevic [17] for the classic framework and overview of early work. More recent contributions include the concept of external (situational) relevance Schamber et al. [18]. Research on selection decision in literature search focused on the ability of users to predict the relevance of documents based on the documents' summaries. For example, Park [16] studied the criteria employed by 10 academic users who were asked to make a selection decision when presented with lists of bibliographic citations. Park categorized user-based characteristics of citation selections as internal, external and problem context. Internal context category describes users perception that are linked at the citation level, for example users perception of author or journals. The external context presents the context stem from individual's search and current research. And lastly, the problem context illustrates why and how the user employs information to construct and solve the information problem. Barry [1] studied the criteria employed by academics to evaluate the representation and the (printed) full text document that has been retrieved specifically for each user's information need. Barry focused on the categorization of user-defined relevance criteria beyond topicality. Her study indicated that the criteria employed by users included tangible characteristics of documents, the provision of references to other sources of information, subjective qualities, and situational factors.

2.2 Selection Decision in XML Retrieval

Selection decision relies heavily on which elements are presented in the summary. XML allows the retrieval and presentation of any individual element in the summary. The presentation of structural text retrieval results is still an open question [10]. Previous user studies have shown the benefits of using XML markup in the retrieval and subsequently information access. Larsen et al. [11] studied whether making elements retrievable is worth the added effort. They found that users find elements useful for their searching tasks, and that they find a lot of the relevant information in specific elements rather than full documents. Betsi et al. [2] found that users liked the idea of being able to gain access directly to the document parts that they were interested in, however, expected the retrieved components to be accompanied by the documents that contain them. Users in this study felt rather uncertain if elements with no contextual information were retrieved. Malik et al. [14] investigated users' behaviors while interacting with XML documents. A result from this study showed that users also appreciated the presentation of XML document structure which is providing context. In terms of elements presented in the summary, only title and authors of documents were displayed as elements summaries in this experiment. As a result, 30 out of 88 test persons in their study commented on the insufficient clues for making a selection decision.

2.3 Selection Decision in Archival Access

Research in users interacting with online finding aids is still in its infancy. Duff and Stoyanova [6] studied elements that were important for users who were looking for archival materials for their research. The following elements were considered to be important: title, information about the creator of the records, call number, scope and content, summary information about content of finding aids, notes of a finding aid, the availability of the finding aids, extent of the material/related records, and types of material/physical description. Since the study mainly focused on the archival display features, they did not elaborate further on the relevance criteria such as user's previous experience and knowledge, sources of information within the environment, and so on.

Duff and Johnson [4] interviewed ten historians focusing on their information seeking behavior. They reported that the historians closely examined finding aids in order to better acquire the sense of the whole collection. They also found that many historians appreciated the addition of summary information about the content of finding aids. This information helped them in their relevance judgments of possible search results. Duff and Johnson [5] studied how genealogists search for information in the archives. Genealogists seek records that contain information about names of people, which might be located in different records. Both studies emphasized the importance of showing the relationships between records (context) and having an overview of the records.

Presenting a list of finding aid elements as a summary has remained a popular method of presenting search results. However, there was no agreement on which elements can be used as summary of the finding aids systems [9]. This can cause a problem since elements used as summary in the search results can vary significantly from one finding aid system to another. For the users, the inconsistency can be very confusing once they interact with several finding aid systems. Presenting why a hit is relevant is strongly related to what to retrieve for the summary. Lee [12, 13] conducted usability studies at the Online Archive of California [OAC, 15], comparing two types of summaries (to which she referred in her study as citations). Long format citation presented title, contributing institution, description (from abstract) and search terms found in information. Short format citation was a Google-like format which presents title and search term. Many of her users preferred long format citation over short format citation, since the long format present more context of the whole collection.

Context of the whole collection was also an issue reported by Fachry et al. [8] where they conducted a study focused

on the effects of presenting context of the whole collection in the hit-list. They conducted a user study were they compare three system: a system that would return the whole fonds[1] (collection level), a system that only returns the individual archival materials (item level) and a system that returns archival material in context (individual items grouped within the same collection). In the first and second systems, the context was omitted, and using this comparison they examined the effects of presenting context in the hit-list. Although the user study showed that the archival material in context system was not optimal, the users had a preference for the third system. The users liked the concept of retrieving archival material in their original context, with users indicating that the system assisted them in assessing relevancy, navigation and direct access to relevant parts of the finding aids.

3. METHODOLOGY

In this section, we discuss the methodology of the user study, specifically we reason our choices of test persons, tasks, summaries, and protocol of the experiment.

3.1 Test persons

The target population of the study included test persons who were novice and expert in searching for archival materials. Although we could elicit more detailed feedback from expert test persons, in this study we also recruited novice test persons, who had no or little experience with archives. They represent a large potential user population for online historical search. In terms of individual differences of test persons, we carefully registered the domain knowledge and archival experiences.

3.2 Tasks

Another important consideration in our study was the tasks. We focus on locating the archival collections of relevance to a given task. In our case, a very specific task as looking for a specific folder number which has a certain topic would be less appropriate. Our interest is in the step before choosing a specific item, where users are presented with a list of results. We prepared four different simulated tasks. The tasks were designed based on the following considerations:

- Tasks were open-ended, requiring test persons to read more than a single summary in order to complete a task.
- The complexity of the tasks was controlled in a way that they were highly similar.
- Each task included background motivation for the search and sufficient information to decide upon the relevance of the viewed summary.

Simulated tasks are presented in the Appendix A. An example of a simulated task is the following:

> *You are interested in the history of slavery in the 18th and 19th centuries. For your history assignment, you are planning to write an essay about anti-slavery movement of that period. To get data for your essay, you are doing research about people who were involved in the anti-slavery movement, who they were and in what way they promoted the anti-slavery movement. Using the digital inventory of the OAC, you would like to check out which archives contain interesting pieces for your research. Depending upon these findings, you should assess whether to visit the archives for your research is worth your time and effort.*

Each test person was assigned two tasks by the experimenters. The order of presentation of the tasks was rotated across test persons. For each task, the test person had to inspect a list of result summaries and decide whether they would view or not to view the result.

3.3 Summaries

In order to operationalize our research questions, we needed "ideal" summaries that contain all potentially useful information. We adopt the summaries used in Online Archive of California [OAC, 15] because they combine both static and query-biased elements in their hit-list. Figure 1 shows a response of OAC finding aid system in relation to the query "Golden Gate Bridge."

We selected ten summaries for each search task. All summaries were prepared by the experimenters. The selection of summaries were based on the following category:

- The selected summaries had a variety of relevance degree to the search task.
- There was a variety of creators. Creators included persons or corporations.
- All summaries were of fonds-level collections that may or may not include series.

Each summary consisted of the following elements:

1. Collection Title, containing the Creator and Title elements
2. Contributing Institution, containing the Repository element
3. Collection Dates, containing the Dates element
4. Items Online, containing the availability and the amount of items online
5. Summary, containing the Abstract element. To avoid confusion between summary element and summary as a whole, the OAC's summary element is referred to as abstract element in this paper.
6. Search term in context. Query-biased summaries/snippets where test persons could see the sentences in which the query terms appeared in the finding aids.

Another methodological consideration was whether to use paper (printed) or digital summary. Summaries printed on paper were chosen rather than digital summaries because:

- A paper summary was an appropriate and sufficient medium to answer our research questions. We were interested to know the contribution of each elements

[1]An archival fonds is all material produced and/or accumulated and used by a person, family or organization over time.

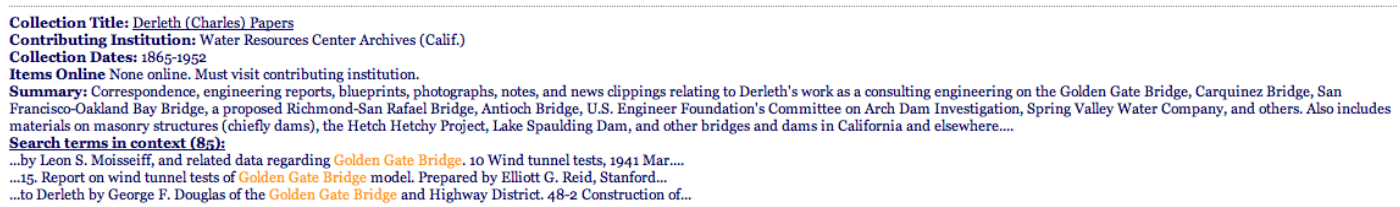


Figure 1: An archival finding aids summary from OAC site (image captured in May 2009)

in test persons' selection decision. Paper summary allowed test persons to easily mark which elements helped them in the selection decision and make notes on how the information in each element helped them in selection decision.

- Paper summary provided ready-transcribed data, the text from test persons' notes can directly be analyzed.

3.4 Protocol

The experiment was designed as follows:

1. Introduction to the experiment and training session.
2. Pre-experimental session in order to collect demographic data of the test persons.
3. Search session I: Judging Summaries. Test person performs the first simulated task, and reviews ten summaries. For each summary, test persons were instructed to:
   (a) examine the summary;
   (b) highlight any portion of the summary that prompted a reaction to pursue the full finding aid;
   (c) for each highlighted portion, comment on the reason to highlight the portion;
   (d) underline any portion of the summary that prompted a reaction not to pursue the full finding aid;
   (e) for each underlined portion, comment on the reason to underline the portion;
   (f) judge the summary as a whole, decide whether to view the full finding aid or not; and
   (g) comment on the reason of the selection decision.
4. Search session II: same as step 2 with a different simulated task.

4. EXPERIMENTAL RESULTS

In this section, we discuss the results of our experiment: the demographics, the elements of the summaries that prompted a positive or negative selection decision, and the motivation behind the choices.

4.1 Demographics data

The total number of 18 test persons (11 male, and 7 female) participated in this study, aged 28–57. All but 5 test persons hold degrees beyond the college (university) level. All test persons were computer-literate with computer experience between 5–15 years. This minimized the possibilities for test persons to find difficulties due to unfamiliarity with common aspects of online navigation.

It is important to emphasize that test persons for this study were carefully registered in terms of their experience with archives. Twelve test persons were recruited from the archive and they all had substantial experience with archives. Ten of the 12 test persons received archival education or training. Out of these 12 test persons, 6 were archivists, 4 were reading room assistants, 1 was a senior adviser and 1 was an ICT manager in an archive. In addition, a thirteenth test person was an amateur-genealogist.

In terms of test persons' experience with archives, 14 test persons had previously conducted historical research (this includes all test persons who were recruited from the archives). When asked about test persons familiarity with archival terminologies, 15 were familiar with archival terminologies in English (this includes all test persons who had conducted historical research). Accordingly, all test persons who were familiar with archival terminologies had visited an archival institution and consulted archival finding aids. However, only 14 of them have visited an archive's site and consulted online finding aids. Since we were using summaries from the OAC, we asked the test persons if they had ever visited the website of OAC. We found out that only 2 test persons had previously visited the OAC website.

4.2 Selection Decision

Table 1 presents test persons' decision in terms of the number of finding aids that they would like to view or not. First, we look at test persons' selection decisions over all tasks. Nine test persons performed simulated task 1, 9 persons did simulated task 2, 10 persons did simulated task 3 and 8 person did simulated task 4. Thus, in total, 36 search sessions were conducted. A total of 360 summaries were examined, since each test person conducted 2 tasks and each task consisted of 10 summaries.

First, we count the number of positive or negative selection decisions based on the test persons' decision to view or not to view a finding aids (see Section 3.4, protocol item 3f). Of the total summaries, test persons decided to view 196 finding aids and not to view 164 finding aids. Thus for each task, a test person decided to view and thus select on average 5.44 finding aids and not to view 4.56 finding aids. The relatively balanced number of view and not view decisions gave us enough feedback to investigate further on processes involved in arriving at a "view" and "not view" selection decision.

Next, we broke down the tasks, and we look at test persons selection decision per task. For each task, did the test persons make the same selection decision? In other words, was there consistency between view or not view decision for each summary within the tasks? For each task, there were 10 cases representing 10 summaries presented to the test persons, 2 categories either 1 (for positive decision) or 0 (for

Table 1: Users' selection decision per task

	Number of Search Sessions	Number of Summaries	Views #	Views %	No views #	No views %	Agreement
Task 1	9	90	43	48	47	52	0.69
Task 2	9	90	49	54	41	46	0.71
Task 3	10	100	56	56	44	44	0.55
Task 4	8	80	48	60	32	40	0.61
Total	36	360	196	54	164	46	

negative decision), and a variety number of raters depending on the number of test person that performed the search task (Task 1=9, Task 2=9, Task 3=10 and Task 4=8). Looking at the agreement for each task using the Kappa statistic [3], the consistency between test persons was substantial for task 1, task 2 and task 4 with K=0.69, K=0.71 and K=0.61, respectively. A moderate consistency was shown for task 3 with K=0.55.

4.2.1 *Elements contributing to selection decision*

We now focus on processes involved in arriving at a "view" and "not view" selection decision. When presented with summaries of results in a hit-list, what information in the summary trigger a positive selection decision to view a result? Table 2 presents elements of a summary contributed to test persons' decision to view a finding aid which we gathered from elements that were highlighted by test persons when they decided to view a finding aid (see Section 3.4, protocol item 3b). In total, the test persons highlighted 443 elements. On average, for each summary, test persons highlighted 2.26 elements (443 highlighted elements/196 view decisions). The elements abstract, title and snippets came first, second and third, followed by elements dates, item online and contributing institution. Out of all finding aids that the test persons viewed (n=196), abstract element contributed the most to a view decision. Test persons highlighted 147 abstract elements (or 75% of what was viewed). Following the abstract element were title and snippets elements with 103 titles (or 53% of what was viewed) and 101 snippets (or 52% of what was viewed). Furthermore, test persons highlighted 54 date elements (or 28% of what was viewed), 35 item online elements (or 18% of what was viewed) and 3 contributing institution (or 2% of what was viewed).

When presented with summaries of results in a hit-list, what information in the summary trigger a negative selection decision not to view a result? Table 3 presents elements of a summary contributed to test persons' decision not to view a finding aid which we gathered from elements that were underlined by test persons when they not viewed a finding aid (see Section 3.4, protocol item 3d). In total, the test persons underlined 241 elements. On average, for each summary, test persons underlined 1.47 elements (241 underlined elements/164 not view decisions). The elements abstract, title and date elements came first, second and third, respectively, followed by item online element, snippets and contributing institution. As with the view decision, out of all finding aids that were regarded as irrelevant by the test persons (n=164), the abstract element contributed the most to a not view decision. Test persons underlined 96 abstracts (or 59% of what was not viewed). Following the abstract element were the dates and item online elements with user underlined 58 title elements (or 35% of what was not viewed) and 36 date elements (or 22% of what was not viewed). Furthermore, test persons underlined 34 item online elements (or 21% of what was not viewed), 16 snippets (or 10% of what was not viewed) and 1 contributing institution element (or 1% of what was not viewed).

Comparing each elements marked (either highlighted or underlined) by the test persons in Tables 2 and 3, there were two interesting findings. First, we can see that the number of elements marked were higher when test persons decided to view a finding aid (n=2.26 elements per summary) compare to when test persons decided not to view a finding aid (n=1.47 elements per summary).

Another interesting finding was the number of snippets marked. We can see that when test persons decided to view a finding aid, 52% snippets were highlighted. While when test persons decided not to view a finding aid, only 10% snippets were underlined. A plausible reason why this happens is that test persons were first reading the general overview of the finding aid (title and/or abstract element). Once they thought the finding aid was relevant, test persons then went further to the snippets and highlighted the terms they found there. In some cases, if test persons did not find the title or abstract elements relevant to their search task, they did not go further to see the snippets part of the summary. It is also worth noting that many of the snippets were incomplete and too short to judge the relevancy of the document. The snippets were useful to see that the query terms appeared in the finding aids, but the information in the snippets was too little to interpret. This could explain why snippets did not contribute much to the test persons' negative selection decisions.

4.3 Motivation for the selection decisions

We go further on how the individual elements contributed to selection decisions. To answer this question, we focus on how test persons interpreted and used each elements presented as summaries.

4.3.1 *Users' assessment of the elements*

During the summary judgment phase in the experiment, we asked test persons to comment on the reason that makes them highlight/underline the elements of the summary (see Section 3.4, protocol items 3c and 3e). The result presented in the following is categorized per elements presented in the summary. Our interpretation of factors contributing to selection decision is presented *in italic* and test persons comments are presented "within brackets."

Title In many cases, the title provided *topical relevance*: "Collection title indicates relevance, even without reading the summary I know that there will be a LOT!" On the other hand, the title can also be a reason to reject a summary due

Table 2: Elements trigger a "view" selection decision

	Title		Institution		Dates		Item Online		Abstract		Snippet	
	#	%	#	%	#	%	#	%	#	%	#	%
Task 1	24	56	0	0	6	14	9	21	34	79	22	51
Task 2	36	73	2	4	18	37	22	45	33	67	19	39
Task 3	25	45	0	0	12	21	4	7	46	82	27	48
Task 4	18	38	1	2	18	38	0	0	34	71	33	69
Total	103	53	3	2	54	28	35	18	147	75	101	52

Table 3: Elements trigger a "not view" selection decision

	Title		Institution		Dates		Item Online		Abstract		Snippet	
	#	%	#	%	#	%	#	%	#	%	#	%
Task 1	11	23	0	0	3	6	10	21	38	81	3	6
Task 2	18	44	1	2	13	32	7	17	12	29	4	10
Task 3	10	23	0	0	15	34	11	25	25	57	2	5
Task 4	19	59	0	0	5	16	6	19	21	66	7	22
Total	58	35	1	1	36	22	34	21	96	59	16	10

to its irrelevancy to the information need: "Title implies that the pictures are about the camp and not about the buildings." The *readability* was an important reason for an element not to trigger a selection decision: "The title does not tell me anything." In this case, the title could be too short or mentioned the creator of the collection who was unfamiliar to the test persons. When this was the case, test persons read the other elements or immediately rejected the summary. The title also provided information about the *type of item* available in the collection: "Scrapbook with only pictures of earthquake." For several test persons, a scrapbook was not relevant to them. As mentioned by one test person: "I need written material for my essay, because I do not want to write about the interpretation of the images."

We could also see that the title gave information about the *author* who collected the documents: "This archive was initiated by a state commission, who should do a very thorough work." In this case, the title gave a positive indication since the author seemed to collect reliable materials. While in another case, the author indication in the title could also be an indication to reject a summary: "A very small collection of snapshot made by an unknown individual: just do not know what the photographs are about and are likely not very specific" Another criteria interpreted from the title was the *specificity or broadness* of the collection: "This collection is too broad in subject matter." In this case, the collection was rejected due to its broadness of the topic area.

Dates The dates were important to show the *time period*: "This is excellent for visiting ... over 16 years on the California proposition." Another interesting finding was the dates gave interpretation of *what the collection contains*: "Long period, probably a lot of material about other topics." In this case, since the date period was too long, he thought that the collection would be too broad and contained many other topics (including not relevant topics). The dates also gave indication of the *recency* of the collection: "The dates are too recent."

Items online The availability of online item gave indication of *effort* that test persons needed to spend: "This archive contains online items, which means I can quickly look at the material first before I decide if I need to visit the institute to view the entire collection." The online information also gave indication of *time* that was needed to see the whole collection: "Too much! (referring to 7,000 pages of text)." In this case, the user decided to reject the summary because the amount of text available gave him a clue that he needed to spend a lot of time to read the collection.

Abstract Abstract provided background information such as the time period covered and a brief history of the organization or person who created the records. Abstract elements were most frequently by the test persons. The main reason why abstract was important because it provided the *overview of the whole collection* : "Though this is a very broad collection because of its scope on the African American, it does hold valuable information. First of all on the movement in general and secondly about some of the people involved." Another example showed how a user interpreted the overview of the collection through the abstract. In this case, the test persons rejected the summary because his information need only appeared in some part of the collection: "The summary mentioned that archive focuses mainly on legislation which are not the focus of my research. Though it states includes "some" material on education project, it is not enough for me to view it."

The *type of document* was also shown in abstract: "The summary does not say what these "letters" are about? Although they pertain to the Gold Rush, it is unclear to me whether these are personal letters containing interesting fact about gold seeker's life style or about something else. Too vague." Not only in title element, abstract element also showed indication of *specificity/broadness* of a collection: "The summary was very specific and detailed. It tells me exactly what I can expect from this letter."

From the summary, test persons also predicted the *time and effort* they needed to spend in reading the records: "It is interesting, 100 relevant pages, it is not online, but I know it is one item (a book with 100 pages), it would depend on time." An example where a test person rejected a summary due to *time/effort*: "Description does not indicate that research would be profitable compared to time consumption."

Another selection decision factor is *novelty/new information*: "Personal archive and different type of media, not only governmental archives." In this case, test persons decided to view the finding aid because it could potentially give a new

information to his research. Another important point was the *originality* of the records: "Letters are primary source." In many cases, test persons would like to see the original source of document, not the result that other people have produced: "Scientific info, I would prefer to read original document. Books I can read in the library, I do not need to go to an archive."

Authorship was another factor why abstract was important. Information in the abstract explained the authorship of the records: "The letters might contains personal experience since he wrote to his mother. I expect that the son is writing a long letter with a lot of information." When a record was authored not by the source, the record could be rejected by the test persons: "It is the son's interpretation of his father's life. Probably biased." Especially in one of the tasks when test persons' task was to explore the life of the gold seekers, the originality of the document and the authorship were important selection decision factors. Another selection decision factor was *the types of item*: "Correspondence is interesting. It may give his (Atkin's) personal points of view." This factor was also related to the authorship of the records.

Snippets Snippets were mainly used to indicate *relevance*: "The terms education and tobacco trigger me to have a look." Test persons also looked at the *specificity* of the item in the snippets: "The search term indicate that this professor in history did research himself in this topic..." In this case, the specific information of the item presented in the snippets, was helpful because it provided detailed information that was not shown in other elements. Another important selection decision factor was the test persons' *ability to understand* the snippets: "This snippet does not tell me anything." Often the snippets were too short or repeated in previous elements in the summary. Unavoidably, the length of the snippets influenced our result in terms of the importance of the snippets in supporting test persons to make a selection decision.

5. DISCUSSION AND CONCLUSIONS

In this paper, we investigated the impact of the result summaries on user's decision to either click or not. We researched this question for a special document genre, archival finding aids, where results have a complex document structure and currently available systems experiment with structured summaries having both static and query-biased elements. Static summary elements contains contextual information about the entire collection. Query-biased summary snippets are selectively extracted on the basis of its relation to the searcher's query. The summaries used for our study consist of five static elements (collection title, contributing institution, collection date, items online, and abstract) and multiple query-biased elements (showing keywords in context) per result.

Our first research question was: What information in the summaries triggers a positive selection decision to view a result and what information triggers a negative selection decision? In general, test persons made a selection decision in two steps. First step of selection decision was assessing the general overview of the finding aid to understand what the collection was about. They assessed this by assessing the static elements: title and abstract of the document. Both in the case of a positive decision to view a document, as well as for a negative decision to skip a document, the title and the abstract elements triggered the selection decision the most. Second step of selection decision was assessing the item description which describes the individual document. Test persons assessed this by looking at query-biased summary/snippets. Looking at the elements that contributed to view selection decision, the elements abstract, title and snippets came first, second and third, followed by dates, item online and contributing institution. Looking at the elements contributed to not view decision, the elements abstract, title and dates came first, second and third, followed by item online, snippets and contributing institution.

Our second research question was: Why and how does this information influence the decision to click or not to click? Each element contributed in the selection decision in different ways. Title element indicated relevance, type of item, author information and specificity or broadness of the collection. Dates element showed information of time period, what the collection contained and the recency of the collection. Online element gave indication of effort that test persons need to spend, and time that was needed to read the collection. Abstract element was marked the highest by the test persons which means the summary was the most useful element in selection decision. Abstract element presented the overview of the whole collection, the type of document, specificity/broadness of a collection, time/effort the test persons need to spend, novelty of the collection/new information, originality of the records, authorship, and the types of collection. Snippets provided indication of relevance and specificity of the item. For the title and the snippets elements, we also found that test persons' ability to understand the information played an important role in test persons' selection decision.

Finally, we go back to the overall aim of this paper: What is the impacts of the result summaries on the users' decision to click or not to click? We concluded that contextual information about the document undoubtedly played an important role in supporting test persons in making a selection decision. For both view and not view decisions, test persons needed sufficient contextual information. Often this information was found in the title and abstract elements. This may be a result of the completeness and coherence of the information in these elements. A title element, although it is short, is a complete sentence and that affects the readability of the element. An abstract element, as compared to the other elements, is by far more complete and coherent and presents what the document is about. Only when the test persons could fully comprehend the information in the query-biased snippets, snippets were used to assess relevancy of the material and to see the detailed description of the document. Length of the information presented in the element also played a clear role. A whole paragraph (as in abstract) triggered a decision more frequently than a short sentence (as in title) or an incomplete sentence (as in query-biased snippet). Further research should answer how much information is needed for contextualizing the results, by studying the length of elements and the importance of the presence of the shown, but not marked elements, in this and other document genres.

Acknowledgments We thank the test persons for donating their time. This research is supported by the Netherlands Organization for Scientific Research (NWO) under grant # 639.072.601.

REFERENCES

[1] C. L. Barry. User-defined relevance criteria: an exploratory study. *Journal of the American Society for Information Science*, 45(3):149–159, 1994.

[2] S. Betsi, M. Lalmas, A. Tombros, and T.Tsikrika. User expectations from XML element retrieval. In *In Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 611–612, 2006.

[3] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measures*, 20:37–46, 1960.

[4] W. M. Duff and C. A. Johnson. Accidently found on purpose: Information seeking behavior of historians in archives. *Library Quarterly*, 72:472–496, 2002.

[5] W. M. Duff and C. A. Johnson. Where is the list with all the names? Information-seeking behavior of genealogists. *The American archivist*, 66:79–95, 2003.

[6] W. M. Duff and P. Stoyanova. Transforming the crazy quilt: Archival displays from a user's point of view. *Archivaria*, 45:44–79, 1998.

[7] EAD. Encoded archival description version 2002, 2002. `http://www.loc.gov/ead/`.

[8] K. N. Fachry, J. Kamps, and J. Zhang. Access to archival material in context. In *Proceedings of the 2nd Symposium on Information Interaction in Context (IIiX 2008)*, pages 102–109. ACM Press, New York NY, USA, 2008.

[9] N. G. Huffman. Search features and other characteristics of XML retrieval systems for EAD finding aids: A content analysis. Master's thesis, School of Information and Library Science, University of North Carolina, April 2008.

[10] J. Kamps. Presenting structured text retrieval results. In *Encyclopedia of Database Systems (EDS)*. Springer-Verlag, Heidelberg, 2009.

[11] B. Larsen, A. Tombros, and S. Malik. Is xml retrieval meaningful to users? searcher preferences for full documents vs. elements. In *Proceedings of the 29th ACM SIGIR Conference*, pages 663–664, 2006.

[12] J. Lee. OAC first round usability test findings. *OAC redesign project*, September 2008. `http://www.cdlib.org/inside/projects/oac/oacredesign.html`.

[13] J. Lee. OAC second round usability test findings. *OAC redesign project*, June 2009. `http://www.cdlib.org/inside/projects/oac/oacredesign.html`.

[14] S. Malik, C.-P. Klas, N. Fuhr, B. Larsen, and A. Tombros. Designing a user interface for interactive retrieval of structured documents: Lessons learned from the INEX interactive track. In *10th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 291–302, 2006.

[15] OAC. Online Archives of California, 2009. `http://www.oac.cdlib.org/`.

[16] T. Park. The nature of relevance in information retrieval: an empirical study. *Library Quarterly*, 63:318–351, 1993.

[17] T. Saracevic. Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science*, 26(6):321–343, 1975.

[18] L. Schamber, M. Eisenberg, and M. Nilan. A reexamination of relevance: toward a dynamic, situational definition. *Information Processing and Management*, 26(6):755–776, 1990.

[19] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10, 1998.

[20] A. Turpin, F. Scholer, K. Jarvelin, M. Wu, and J. S. Culpepper. Including summaries in system evaluation. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 508–515. ACM Press, New York NY, USA, 2009.

[21] XML. Extensible markup language (XML) 1.0 (fourth edition), 2006. `http://www.w3.org/TR/xml/`.

APPENDIX

A. SIMULATED TASKS

Task 1 You have been asked to organize an activity as part of tobacco education program for high schools students. To get inspiration, you are doing research on previous activities that attempted to give tobacco education for school-age children. For example, you want to know what organizations were actively promoting tobacco education for school-age children, what purposes they had, and what anti-tobacco education projects and activities they implemented.

Task 2 You are writing an article about the damage of the 1906 San Francisco earthquake on buildings of San Francisco. As you know, the earthquake and resulting fire is remembered as one of the worst natural disasters in the history of the United States. To get data for your article, you want to know which buildings the earthquake damaged and to find photographs of the damaged buildings.

Task 3 You are interested in gold rush topic in the California, which happened in the 19th century. For your history assignment, you are planning to write an essay about gold rush at that time. To get some data for your essay, you are doing research about people who came to California as gold seeker, who they were and how their life was as gold seekers during the gold rush period.

Task 4 You are interested in the history of slavery in the 18th and 19th centuries. For your history assignment, you are planning to write an essay about anti-slavery movement of that period. To get data for your essay, you are doing research about people who were involved in the anti-slavery movement, who they were and in what way they promoted the anti-slavery movement.

Search Request (for all tasks) Using the digital inventory of the OAC, you would like to check out which archives contain interesting pieces for your research. Depending upon these findings, you should assess whether to visit the archives for your research is worth your time and effort.

Learning to Merge Search Results for Efficient Distributed Information Retrieval

Kien-Tsoi T. E. Tjin-Kam-Jet
University of Twente, The Netherlands

Djoerd Hiemstra
University of Twente, The Netherlands

ABSTRACT

Merging search results from different servers is a major problem in Distributed Information Retrieval. We used Regression-SVM and Ranking-SVM which would learn a function that merges results based on information that is readily available: i.e. the ranks, titles, summaries and URLs contained in the results pages. By not downloading additional information, such as the full document, we decrease bandwidth usage. CORI and Round Robin merging were used as our baselines; surprisingly, our results show that the SVM-methods do not improve over those baselines.

Keywords

Distributed information retrieval, results merging, interleaving, round robin, learning to rank, meta-search, federated-search, collection fusion.

1. INTRODUCTION

Centralized search is limited by its inability to search through the *deep web*—pages accessible only after querying an HTML form, since web crawlers lack the intelligence to adequately fill in and submit such forms. Another drawback is that the index needs to be maintained and updated to cope with both content change and Web growth [1].

With Deep Web content already residing in searchable databases, and in the expectation that the Web will continue its enormous growth, a promising search paradigm is DIR (Distributed Information Retrieval) [3]. A DIR system contains at least one broker and multiple servers, each indexing its own document collection. The broker serves as a mediator between the user and the servers. The user sends a query to the broker, which subsequently selects the servers most capable of adequately answering the query, and forwards the query to the selected servers. Each server then retrieves its most relevant documents and sends a ranked list of results back to the broker, which merges these into one results list and presents it to the user. Generally, although the broker only controls the way in which the servers are selected and the way their results are merged, it has no control over the internal functioning of any server.

DIR is a well-established research area with three main areas of interest: server description, server selection, and results merging [3]. A server description is often an excerpt of a server's index and it is used to estimate the number of different words and word frequencies of the server [4, 11]. In this way, server selection is done by treating each excerpt as one very large document, and subsequently applying standard IR technology to rank and select the top N servers. Most existing result-merging methods require the server to supply a document score—otherwise an estimate of the score is used. These scores are then adapted so that inter-server document scores can be compared and ranked. However, in practice, document scores are hardly ever provided by search servers, or if they are, they cannot be trusted.

In this paper, we propose the use of information from search result snippets that search servers typically provide: the document title, its url, and a dynamically generated document summary containing the matching query terms. Unlike in previous work, our broker does not have any excerpt of any server's index, nor does it require document scores to be supplied along with the server's results. Therefore, we apply methods that neither rely on estimated indices and document scores, nor on the download of any additional information, such as the full document. We use SVM [16] (Support Vector Machine) to train a function for merging results based only on evidence contained in the results pages received from the servers. In addition, the benefit of not downloading any additional information is decreased bandwidth usage.

Outline of paper: Section 2 summarizes key literature about results merging. Our experiment testbed is explained in Section 3. Section 4 presents our merging approach, and the evaluation is discussed in Section 5. Section 6 presents and discusses our results, and Section 7 gives our conclusion.

2. RELATED WORK

2.1 CORI

CORI [3, 5] has been used by many researchers [7, 9, 8, 13] as a baseline for server selection and results merging. Query Based Sampling (QBS) [4] is often used to obtain the server descriptions needed to run the CORI server selection algorithm which ranks the servers based on the belief-score

of observing the query's terms in that particular server.

Once the results pages are obtained from the selected servers, the document scores given by the distinct servers are normalized and weighted as follows:

$$w = 1 + 0.4 * \frac{s - S_{min}}{S_{max} - S_{min}}, \quad (1)$$

$$D' = \frac{D - D_{min}}{D_{max} - D_{min}}, \quad (2)$$

$$D'' = \frac{D' * w}{1.4}. \quad (3)$$

where s is the server's belief score; S_{min} and S_{max} are the highest and lowest belief scores respectively that CORI could potentially assign to a server; D is the document score supplied by the server; D' is the normalized document score; and D'' is the weighted document score.

Note that (2) requires cooperation among servers because D_{max_i} and D_{min_i} must be provided by the server when it returns document rankings. Our goal is *not* to rely on any form of cooperation, because cooperation can be unreliable in multi-party environments. In the absence of cooperation, D_{max_i} is set to the maximum document score returned by the server and D_{min_i} is set to the minimum [13].

2.2 Ranking-SVM

Joachims developed an SVM-type called Ranking-SVM [6], he used it to learn a preferred ranking function from clickthrough data. He argued that clickthrough data can be recorded at very low cost, and that users make a (reasonably) informed choice when clicking on a link, instead of clicking at random. Therefore, clicks are likely to convey some partial ranking information that can be used to learn a ranking function.

For example, if a user clicked on results 3 and 5, the preferred ranking would be: 3,5,1,2,4. In other words, the system made some errors: it should have ranked result 3 ahead of results 1 and 2, and result 5 ahead of results 1, 2 and 4. These five errors, called preference constraints, are deduced from the clicks (plus the ranked list) and serve as the input for the SVM^{light} program that Joachims developed. The input consists of (labeled) document pairs, where one document is preferred over the other. The program tries to learn a ranking function that maximizes the proportion of correctly-ordered pairs of documents (induced by the learned ranking function when compared to the preferred rankings).

2.3 Regression-SVM

Several researchers [7, 10, 12] tackled the results merging problem by learning a regression function that maps server-specific document scores to centralized document scores—centralized scores are derived from a central index that contains many sampled documents from all servers. The motivation behind this approach is that the document scores produced by all servers are usually incomparable.

Inspired by this approach, we decided that, instead of mapping server-specific document scores to centralized document scores, we could use Regression-SVM to learn a function that directly determines the "centralized" score of a document, given its features.

3. TESTBEDS

We used the multi-purpose TREC WT10g [2] collection as a testbed for our experiments. Our experiments require result pages from different servers (each indexing different documents), as well as some server selection mechanisms. The WT10g corpus was not necessarily created for conducting DIR experiments. Therefore, we created two different testbeds containing result pages from different servers. The following subsections describe our testbeds and present several server selection mechanisms.

3.1 Result Page Creation

The PF/Tijah retrieval platform was used to create result pages for which each result has a rank, title, summary and URL. PF/Tijah expects its input to be valid XML. Therefore, the first step was to convert all WT10g data into valid XML. We used a program that: 1) discarded the HTML comments, scripts, and all HTML tags except the title and anchor tags; 2) truncated URLs by removing all '/index...'-endings, such as /index.html; 3) marked 'sentence-boundaries' in such a way as to create sentences of about 40 to 160 characters. This was done for the purpose of sentence ranking, which is used for creating the document summaries [15]; finally, 4) if a document did not have a title, a title was created from the first sentence of the document.

The second step was to re-group the web pages by their IP-address. This resulted in XML documents containing all web pages from a single server, and we refer to these newly created documents as *ip-grouped* documents. We regrouped the web pages because we assumed that the pages that make up a website are highly related to each other and that they most often reside on the same web server. Since the web pages in the original WT10g corpus were randomly distributed over several file chunks, we had to perform this additional step.

The third step is to create servers and populate them with the ip-grouped documents. A simple set of rules was used to create these servers. First, we sorted the ip-grouped documents by their file size. Then we selected the smallest ip-grouped document and assigned it to a server *only* if the server was empty or if the server's new size would not exceed a specified size of X MB. Note that an ip-grouped document bigger than X MB was not split. We created two testbeds for our DIR experiments by setting X to 100MB and 500MB. Splitting the WT10g corpus in chunks of roughly 100MB resulted in 79 servers, whereas splitting in chunks of 500MB resulted in 15 servers.

In the fourth and final step, we created an index for each server, and submitted the queries to the servers to obtain the required result pages. These pages contained a maximum of 50 results, and a number indicating the total number of documents found by that server.

3.2 Server Selection

A user is typically only interested in the first N, say 20, results. This means that querying more than N servers wastes valuable resources. In addition, it is not efficient to query a server that will return no relevant results. Therefore, the broker *must* select a small number of the most promising servers.

A results merging method should produce the best possible merged-rankings given any (possibly very poor) set of selected servers. However, we are still far from that ideal. A random server selection or one based on the server's retrieval performance would probably yield significantly different merged-rankings, even in the case where the identical set of servers were selected, albeit in a different order. A server's retrieval performance can be measured by, for instance (4), the Average Precision (AP) measure [17].

$$\mathrm{AP} = \frac{\sum_{i=1}^{N} \mathrm{precision}(i) * \mathrm{rel}(i)}{reldocs} \ . \qquad (4)$$

where precision(i) is the fraction of relevant documents retrieved up to and including rank i; rel(i) is a binary function producing the value 1 when a document at rank i is relevant and 0 otherwise; and *reldocs* is the number of relevant documents in the document collection for this particular query.

Several server selection strategies are briefly described below.

CORI The CORI server selection algorithm—using the complete (i.e., no QBS) term statistics from each server's index to calculate the CORI-belief score.

Merit A strategy that ranks the servers based on the number of relevant documents in their document collection.

Local-AP A performance-based selection strategy similar to (4), but where *reldocs* refers to the number of relevant documents in the server's document collection.

Global-AP A performance-based selection strategy similar to (4), but where *reldocs* refers to the number of relevant documents in the combined document collection of all servers.

4. MERGING APPROACHES

We implemented two SVM learning methods: Ranking-SVM and Regression-SVM. We used Round Robin (RR) and CORI (which was briefly discussed in Section 2.1) as our merging baselines. However, CORI-merging requires the belief scores produced by the CORI-selection schemes; therefore, whenever we use other selection schemes, RR is our only baseline.

The remainder of this section elaborates on the RR and SVM merging approaches.

4.1 Round Robin

Round Robin merging is the simplest merging method and is defined as follows: given n result lists $L_1, L_2, \ldots, L_n$, take the first result r_1 from each list L_i as the first n results. Then take the second result r_2 from each list as the next n results, and so on. RR merging produces a list: $L_1r_1, L_2r_1, \ldots, L_nr_1$, $L_1r_2, L_2r_2, \ldots, L_nr_2$, $L_1r_3, L_2r_3, \ldots, L_nr_3$, etcetera.

Often, the rank of the results is the only feature used when doing RR merging. However, with information about the relevant document distributions of the servers, i.e. the *server score*, we could first rank the servers. By combining both the server score and the result rank, RR can pick the next best result from the next best server, thereby improving its merging performance.

4.2 Learning

This subsection explains the features and labels of the training data for both SVM approaches, and how we validated our models.

4.2.1 Features

Table 1 lists the features used in our experiments. All features are grouped into some category and each category states the number of features between brackets. For example, the second group (Server rank) has one feature which is the score given by one of the four server selection strategies, whereas the final group (Result's term diversity) has three features telling us something about the diversity of the words and characters contained in a given result. The abbreviations LCS, LWO, and LM denote Longest Common Substring, Longest Word Order, and Language Model respectively. The letters q, t, s, f, p, and u stand for query, title, summary, fqdn (Fully Qualified Domain Name), path, and URL (u = f+p), respectively.

LM(a,b) is a simple language model similarity between a and b: the term-frequency statistics are taken only from the text found in b, and a constant of 0.001 is used for smoothing. We also implemented an LM algorithm that allows partial matching (denoted by LM-p). An example of partial matching is when the query 'chair' matches a piece of text such as 'wheelchairs.com'.

LCS(a,b) detects the greatest unaltered proportion of string a that also appears in exactly the same way in b. LWO(a,b) is almost similar to LCS, but it allows for noise. For example, let a denote the text "using ranking SVM in IR" and let b denote "using Machine Learning techniques for ranking in IR". The LCS similarity between a and b is fairly low (0.4), while the LWO similarity yields a score of 0.8.

For a given server, D_{found} denotes the total number of documents found. D_{min} and D_{max} denote the minimum and maximum number of documents respectively found by the selected servers.

We grouped the features for the purpose of feature selection: when we trained a model, we tried different combinations of the feature groups. Note that the result rank feature was used differently in the two SVM approaches. With the linear rank score, Ranking-SVM performed extremely poorly, while it performed much better with the logistic rank score. For Regression-SVM, the effects of the rank features were the other way around, although the logistic feature was not as dramatic for Regression-SVM as the linear feature was for the Ranking-SVM.

Finally, we also experimented with stemmed and stopped versions of the final six feature groups. In later sections, we will append the suffix '*-ws*' to denote that stemming and stopping were used, and the suffix '*-ns*' to denote that stemming and stopping were not used.

4.2.2 Ranking-SVM

Clicks indicate a preferred ranking that should be learned by the Ranking-SVM algorithm. However, we do not have actual click data, so instead we use the TREC relevance judgments. There are important differences between the two. Clicks are binary and convey relative relevance that is based on superficial information supplied by the search engine (e.g., ranks, titles, summaries, and URLs). WT10g TREC judgments are ternary and convey *absolute* relevance: a team of people have actually read the entire document and

Table 1: List of Features

Result rank (1)
$1 - rank/50$ (for Regression-SVM)
$1 - 1/2 * \log(rank)$ (for Ranking-SVM)
Server rank (1)
the normalized server score
Documents found by server (1)
$(D_{found} - D_{min})$ / $(D_{max} - D_{min})$
Server response (1)
LM: q – top10 server results
Digits (20)
number of [1–4]-digit numbers in $\{q, t, s, f, p\}$
Path (1)
the amount of '/'-characters in p
Language model (4)
LM-p: q – {t, s, f, p}
Longest common substring (4)
LCS: q – {t, s, f, p}
Longest word order (4)
LWO: q – {t, s, f, p}
Result consistency (3)
LM-p: t–s, t–u, s–u
Word statistics (10)
number of words in {q, t, s, f, p}
avg. word length in {q, t, s, f, p}
Result's term diversity (3)
total distinct terms / total terms
most frequent term's frequency / total terms
total non-word characters / total characters

rated it as being irrelevant, relevant, or highly relevant.

Furthermore, the assumption that users scan the ranks sequentially from top to bottom allows us to further assume that a higher ranked document that was not clicked is probably less relevant than a lower, clicked, document. This is not the case with TREC judgments; our retrieved documents were not judged in sequential order, so the standard assumption that unjudged documents are irrelevant might lead to learning a sub-optimal ranking function when treating unjudged results as irrelevant. Therefore, we decided to discard the unjudged results when training an SVM model.

As an example of how we used the TREC judgments to create the preference constraints, consider the following rankings where result 2 is irrelevant, results 1 and 5 are relevant, and result 3 is highly relevant. Discarding the unjudged result, the preferred ranking is: 3,1,5,2. The preference constraints are $3 \succ 1$, $3 \succ 2$, and $5 \succ 2$. For each click, Joachims [6] added random additional constraints that should stabilize the learned ranking. We also added 10% (of the total results being merged) of additional random constraints. In addition, we only chose randomly from the set of results that were less relevant than the 'clicked' document; however, this is impossible if you only have clickthrough data.

Finally, we restricted the ranks at which we "observe" the clicks: we only look for clicks within the top 15% of the rankings. For example, in a page with 50 results, if ranks 7 and 8 are relevant, we only create the preference constraints for the result ranked 7^{th}. This restriction led to a substantial gain in the retrieval performance of the learned ranking function.

4.2.3 Regression-SVM

Using Regression-SVM, we aim to predict the *absolute* rank of a given result. This rank should reflect the gathered knowledge from both the TREC judgments as well as of the servers' rankings. However, the TREC judgment should have a higher impact on the learned ranking function. For instance, if a highly relevant result (according to the TREC judgment) was ranked lowest by some search engine, then we certainly want our learned ranking function to rank that result somewhere near the top.

Just as with our Ranking-SVM approach, we excluded unjudged results from our training data in order to avoid unnecessary noise. We label each training instance simply by the value obtained when deducting its rank from either fifty or one hundred, depending on whether the result was irrelevant or not, respectively. The resulting label ensures that all relevant documents (according to the TREC judgments) are ranked in the top positions, followed by the irrelevant documents. Also, within each class of (relevant or irrelevant) documents, the documents are further ordered based on the original rankings of the search servers.

4.2.4 Validation

Our training data consisted of the result pages for the fifty *odd*-numbered queries, taken from a set of N servers. (The queries were taken from TREC topics 451–550.) The servers were selected using selection strategy S. We also varied the set of features F used for training. During training, we used the default values for the SVM-parameters. Each combination of N, S, and F yields a different training set and thus a (potentially) different model. To validate all these models, and choose the model with the best retrieval performance, we used 25-fold cross-validation.

Each fold determines the set of queries QT that will be used for training, and the set QV that will be used for validation. In particular, we focused our validation on merging results from the top 3, 4, and 5 servers. For instance, for each fold, we validated on (QV, 3, S, F), (QV, 4, S, F), and (QV, 5, S, F), and we recorded the averaged Local-MAP and Global-MAP as that fold's validation score.

After cross-validating, we chose the model with the highest Global-MAP, and the one with the highest Local-MAP; this was done for both Ranking-SVM and Regression-SVM. In other words, we selected a total of four models.

5. EVALUATION

We evaluated the different approaches by measuring their Global-MAP when merging the results of the even-numbered queries of the top N servers, which were selected following one of the available server selection strategies.

To test whether the merging methods were significantly (with $p<0.05$) better than the RR or CORI merging method, we

Table 2: Ranking-SVM weights

	SVM-0		SVM-1
result rank	3.544	result rank	3.325
LWO-*ws*(q, t)	-0.557	LWO-*ns*(q, t)	-0.443
LWO-*ws*(q, s)	0.834	LWO-*ns*(q, s)	1.391
LWO-*ws*(q, f)	0.198	LWO-*ns*(q, f)	0.612
LWO-*ws*(q, p)	-0.898	LWO-*ns*(q, p)	0.162

used a randomization approach [14] with 100,000 random permutations. Our test statistic was the Global-MAP of each merging approach.

6. RESULTS

In this section, we present and discuss the performance of the merging methods: CORI, RR, and the four SVM models that were chosen by cross-validation. We will start by discussing the cross-validation results, after which we will discuss the test results.

6.1 Cross-Validation Results

The Ranking-SVM model which has the highest (cross-validated) Local-MAP was trained on the results pages of the top 3 GAP-selected servers, with the result rank and LWO-*ws* features. We will refer to this model as Ranking-SVM-0.

The Ranking-SVM model with the highest Global-MAP was trained using the results pages of the top 3 GAP-selected servers, with the result rank and LWO-*ns* features. We will refer to this model as Ranking-SVM-1.

The Regression-SVM model which has the highest Local-MAP was trained using the results pages of the top 5 GAP-selected servers, and the following features: result rank, server rank, LCS-*ws*, iz-*ns*. We will refer to this model as Regression-SVM-0.

The Regression-SVM model with the highest Global-MAP was trained using the results pages of the top 3 GAP-selected servers, and the following features: result rank, LCS-*ws*, LM-p-*ns*, iz-*ns*. We will refer to this model as Regression-SVM-1.

The learned feature weights of the models can be seen in Tables 2 and 3. As you can see, the result rank feature is the most important feature.

6.2 Test Results

Figures 1, 2, and 3 show how the Global-MAP changes as the number of selected servers increases. Figures 4, 5, and 6 show how the Precision@10 changes as the number of selected servers increases. There is a figure for each server selection strategy and both collections sizes.

In all six figures, the first row of numbers on the x-axis denotes the number of selected servers, while the second row denotes the average number of relevant documents per query, which is a direct consequence of the server selection strategy.

Keep in mind that we want to select as few servers as possible (e.g., to minimize network traffic and computing time), while at the same time, we want the merging performance to be as high as possible.

Table 3: Regression-SVM weights

	SVM-0		SVM-1
result rank	50.000	result rank	49.142
server rank	0.000		
LCS-*ws*(q, t)	0.000	LCS-*ws*(q, t)	-0.314
LCS-*ws*(q, s)	0.000	LCS-*ws*(q, s)	0.295
LCS-*ws*(q, f)	0.002	LCS-*ws*(q, f)	0.027
LCS-*ws*(q, p)	0.002	LCS-*ws*(q, p)	-0.009
iz-*ns*(q, s)	0.000	iz-*ns*(q, s)	0.062
iz-*ns*(q, f)	0.000	iz-*ns*(q, f)	-0.072
iz-*ns*(q, p)	0.000	iz-*ns*(q, p)	0.044
		LM-*ns*(q, t)	0.132
		LM-*ns*(q, s)	0.139
		LM-*ns*(q, f)	-0.169
		LM-*ns*(q, p)	0.831

When using the LAP and GAP selection strategies, RR is always significantly better than the SVM models. Sometimes, the differences between the SVM models are also significant. When using the CORI selection strategy, from five servers onwards, both CORI and RR are usually significantly better than the SVM models. Keep in mind that when doing multiple comparisons, we would expect some significant differences to actually be false alarms.

6.2.1 CORI selection

Using CORI selection, the retrieval performance of all models is much lower than with any other selection method, as can be seen from the Global-MAP figures as well as the P@10 figures. The performance of both baselines—RR and CORI-merging—is almost indistinguishable.

Compared to LAP-selection, the first few servers selected by CORI-selection contain almost twice as many relevant documents per query on average (the small numbers below the x-axis), yet none of the merging methods seem able to exploit this fact. The extremely poor performance of RR (compared to the other server selection strategies) indicates that CORI-selection often selects servers that return no relevant results at rank one. Furthermore, since no other merging method outperforms RR on this data, it suggests that it is difficult to discriminate between relevant and irrelevant results, at least in this particular set of results.

6.2.2 GAP selection

Using GAP selection, RR clearly outperforms the other merging methods. The margin by which RR outperforms the other models is unexpected, especially since the result's rank seems to be the most important feature for all models (just as for RR), as can be seen in Tables 2 and 3. Note that the range of all feature values lies between one and zero, except for the LM features (of which we have seen values ranging from zero up to five).

7. CONCLUSION

Merging search results from different servers both efficiently and effectively is a major problem in Distributed Information Retrieval.

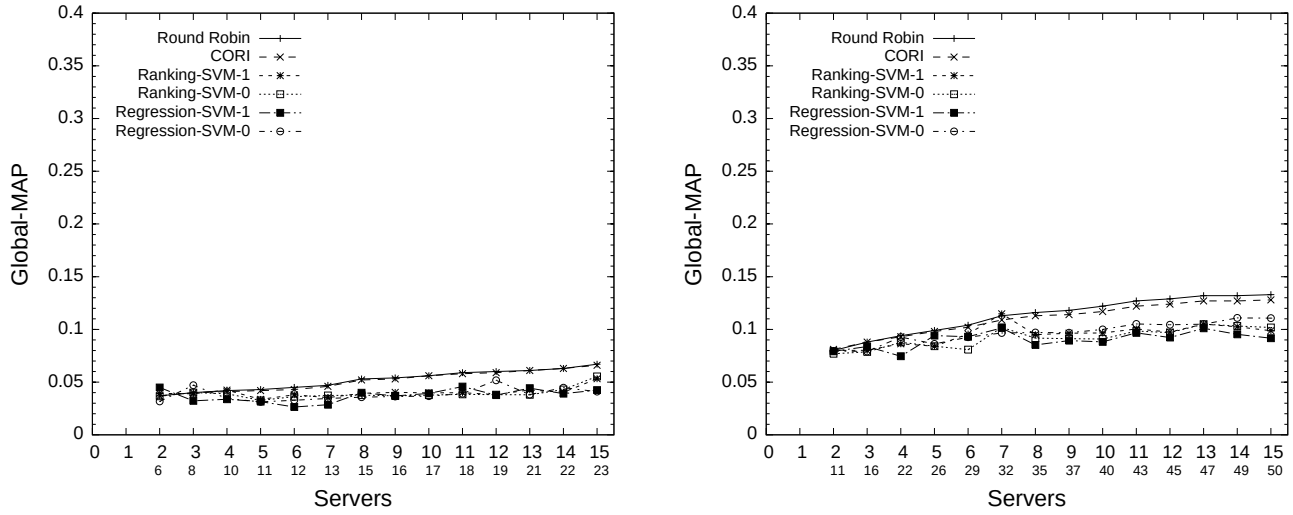


Figure 1: Global-MAP for CORI-selection on the 100MB (left) and 500MB (right) collections

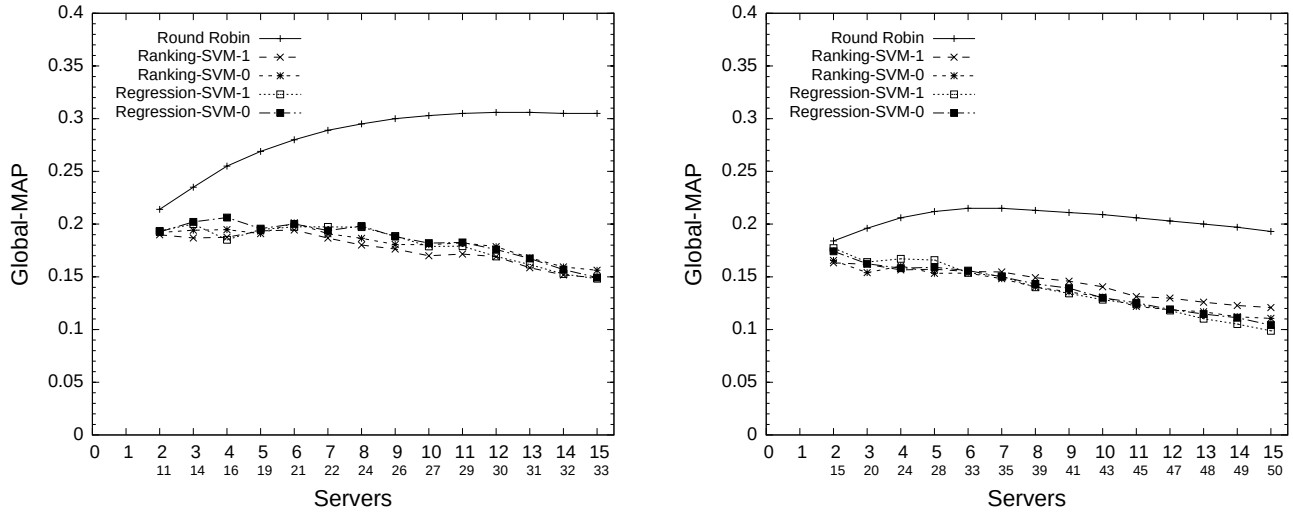


Figure 2: Global-MAP for GAP-selection on the 100MB (left) and 500MB (right) collections

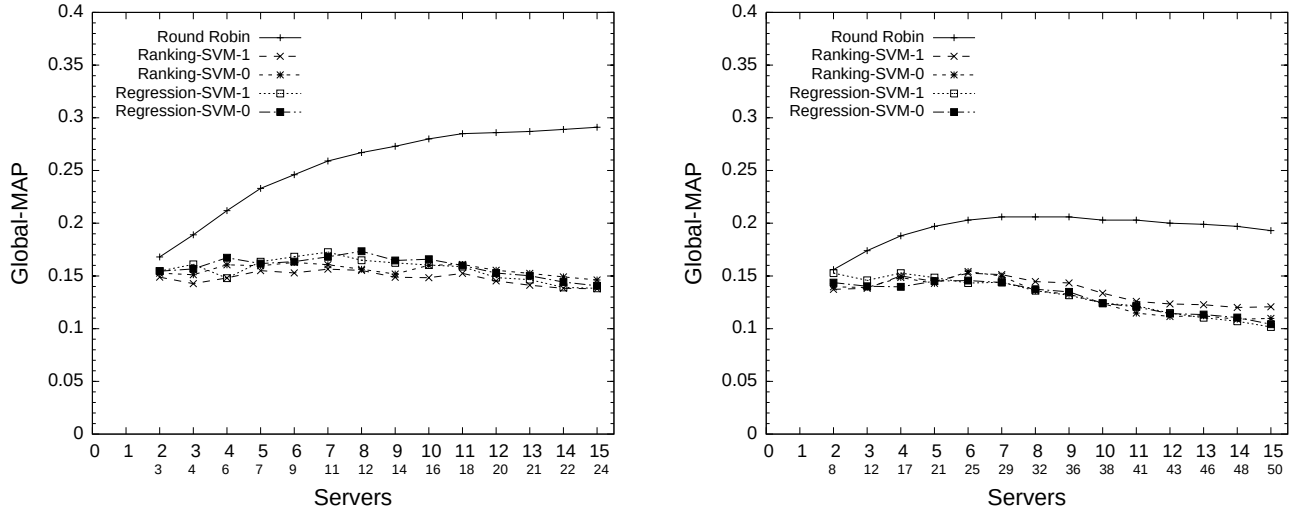


Figure 3: Global-MAP for LAP-selection on the 100MB (left) and 500MB (right) collections

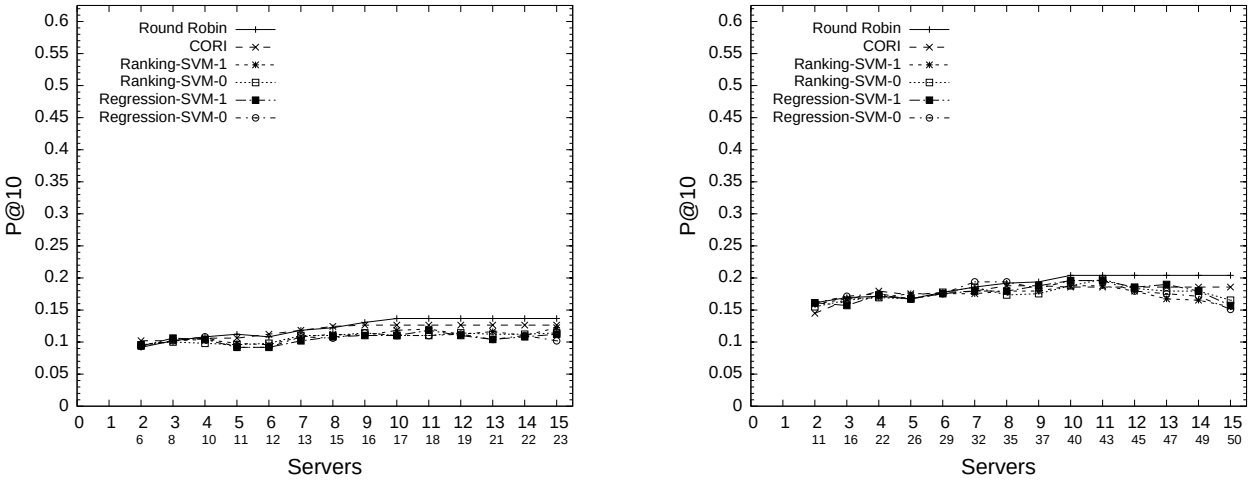


Figure 4: P@10 for CORI-selection on the 100MB (left) and 500MB (right) collections

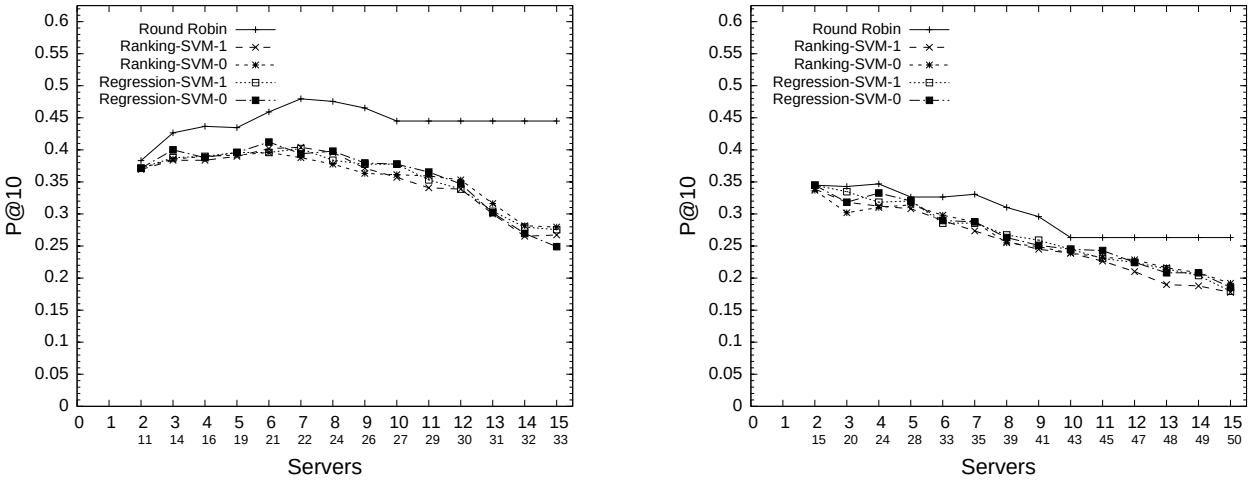


Figure 5: P@10 for GAP-selection on the 100MB (left) and 500MB (right) collections

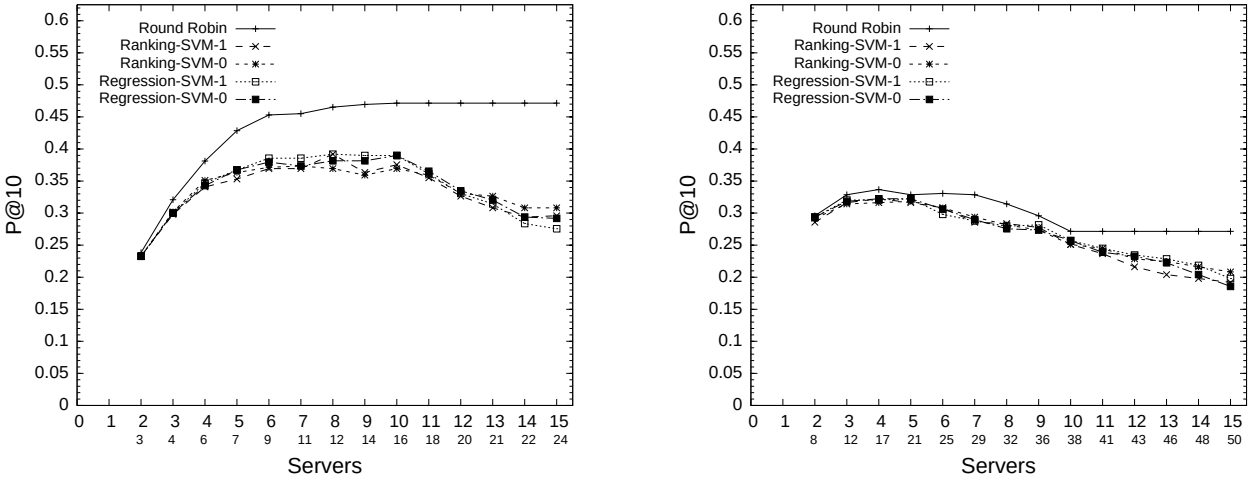


Figure 6: P@10 for LAP-selection on the 100MB (left) and 500MB (right) collections

Our approach avoids the use of document scores and learns a ranking function—using Support Vector Machines—that can merge results based on information that is readily available: i.e. the ranks, titles, summaries and URLs, contained in the result pages. By not downloading additional information, such as the full document, we decrease bandwidth usage.

We have experimented extensively with many different feature combinations to find a good ranking function. We trained a ranking-SVM model that uses pairwise training instances to learn a ranking function, and a regression-SVM model that uses pointwise training instances.

However, our experiments show that the SVM-methods do not improve over the baselines.

8. DISCUSSION

Using Ranking-SVM proved to be very much more sensitive to the type of features used, and the way in which they are preprocessed, as compared to Regression-SVM.

It is disappointing that the SVM approaches were unable to achieve a better performance than Round Robin. One might argue that in real life, no such thing exists as GAP-selection. However, that does not explain why the SVM algorithms apparently learn a mediocre ranking function when trained with exactly these features (i.e., result rank and server rank, as indicated by GAP-selection).

We also experimented with z-normalization for those features that might have a different order of magnitude, depending on the query. Z-normalization works as follows: for a feature f, we compute a new score $s_f^{'} = (s_f - \mu_f)/\sigma_f$, where μ_f is the mean of all values of feature f, and σ_f is the standard deviation of all values of feature f.

Our preliminary results show that this additional normalization does not lead to an improvement of the learned models.

We used a linear kernel for our experiments; therefore, we cannot conclude that our features are insufficient to optimally merge the results. Using a non-linear kernel could lead to a better model. Our motivation for using linear kernels was that Joachims [6] also used linear kernels, and he also used some features that looked similar to the features that we used.

9. REFERENCES

[1] R. A. Baeza-Yates, C. Castillo, F. Junqueira, V. Plachouras, and F. Silvestri. Challenges on distributed web retrieval. In *ICDE*, pages 6–20. IEEE, 2007.

[2] P. Bailey, N. Craswell, and D. Hawking. Engineering a multi-purpose test collection for web retrieval experiments. *Inf. Process. Manage.*, 39(6):853–871, 2003.

[3] J. Callan. *Distributed Information Retrieval*, volume 7 of *The Information Retrieval Series*, chapter Distributed Information Retrieval, pages 127–150. Springer US, 2000.

[4] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Trans. Inf. Syst.*, 19(2):97–130, 2001.

[5] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–28, New York, NY, USA, 1995. ACM.

[6] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142. ACM, 2002.

[7] G. Paltoglou, M. Salampasis, and M. Satratzemi. Results merging algorithm using multiple regression models. In *ECIR 2007, Rome, Italy*, pages 172–184. Springer, 2007.

[8] Y. Rasolofo, F. Abbaci, and J. Savoy. Approaches to collection selection and results merging for distributed information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 191–198, New York, NY, USA, 2001. ACM.

[9] M. Shokouhi. Central-rank-based collection selection in uncooperative distributed information retrieval. In *ECIR 2007, Rome, Italy*, pages 160–172. Springer, 2007.

[10] M. Shokouhi and J. Zobel. Robust result merging using sample-based score estimates. *ACM Trans. Inf. Syst.*, 27(3):1–29, 2009.

[11] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 298–305, New York, NY, USA, 2003. ACM.

[12] L. Si and J. Callan. A semisupervised learning method to merge search engine results. *ACM Trans. Inf. Syst.*, 21(4):457–491, 2003.

[13] L. Si, R. Jin, J. Callan, and P. Ogilvie. A language modeling framework for resource selection and results merging. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 391–397, New York, NY, USA, 2002. ACM.

[14] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In M. J. Silva, A. H. F. Laender, R. A. Baeza-Yates, D. L. McGuinness, B. Olstad, Ø. H. Olsen, and A. O. Falcão, editors, *CIKM*, pages 623–632. ACM, 2007.

[15] A. Turpin, Y. Tsegay, D. Hawking, and H. E. Williams. Fast generation of result snippets in web search. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 127–134, New York, NY, USA, 2007. ACM.

[16] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, New York, Inc., 1995.

[17] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2005.

Simulating Signal and Noise Queries for Score Normalization in Distributed IR

Avi Arampatzis[1] Jaap Kamps[2,3]

[1] Electrical and Computer Engineering, Democritus University of Thrace, Greece
[2] Archives and Information Studies, University of Amsterdam
[3] ISLA, Informatics Institute, University of Amsterdam
avi@ee.duth.gr kamps@uva.nl

ABSTRACT

Score normalization is indispensable in distributed retrieval and fusion or meta-search where merging of result-lists is required. Distributional approaches to score normalization with reference to relevance, such as binary mixture models like the normal-exponential, suffer from lack of universality and troublesome parameter estimation especially under sparse relevance. We develop a new approach which tackles both problems by using aggregate score distributions without reference to relevance, and is suitable for uncooperative engines. The method is based on the assumption that scores produced by engines consist of a signal and a noise component which can both be approximated by submitting well-defined sets of artificial queries to each engine. We evaluate in a standard distributed retrieval testbed and show that the signal-to-noise approach yields better results than other distributional methods.

1. INTRODUCTION

Modern best-match retrieval models calculate some kind of score per collection item which serves as a measure of the degree of relevance to an input request. Scores are used in ranking retrieved items. Their range and distribution varies wildly across different models making them incomparable across different engines [4], even across different requests on the same engine if they are influenced by non-semantic query characteristics, e.g. length. Even most probabilistic models do not calculate the probability of relevance of items directly, but some order-preserving (monotone or isotone) function of it.

The main aim of this paper is to analyse and further develop score distributional approaches to score normalization. Our underlying assumption is that normalization methods that take the shape of the SD into account will be more effective than methods that ignore it. We want to make no assumptions on the search engines generating the scores to be normalized other than that they produce ranked lists sorted by decreasing score. Thus, we treat each engine as a 'black-box' and are interested in approaches based only on observing their input-output characteristics: the queries and resulting score distributions.

*This is an extended abstract of: A. Arampatzis and J. Kamps. A signal-to-noise approach to score normalization. In *Proceedings CIKM 2009*, pages 797–806. ACM Press, New York USA, 2009.

DIR-2010 January 25, 2010, Nijmegen, the Netherlands.

2. SINGLE DISTRIBUTION METHODS

Z-score A standard method for score normalization that takes the SD into account is the Z-SCORE. Scores are normalized, per topic and engine, to the number of standard deviations that they are higher (or lower) than the mean score:

$$\text{Z-SCORE:} \quad s' = \frac{s - \mu}{\delta}$$

where μ is the mean score and δ the standard deviation. Z-SCORE assumes a normal distribution of scores, where the mean would be a meaningful 'neutral' score. As it is well-known, actual SDs are highly skewed.

Aggregate Historical CDF Simplified A recent attempt models aggregate SDs of many requests, on per-engine basis, with single distributions [3] using the historical CDF:[1]

$$\text{HIS:} \quad s' = P(S_{\text{HIS}} \leq s)$$

where $P(S_{\text{HIS}} \leq s)$ is the *cumulative density function* (CDF) of the probability distribution of all scores, and HIS refers to the fact that historical queries are used for aggregating the SD that the random variable S_{HIS} follows. HIS normalizes input scores s to the probability of a historical query scoring at or below s.

3. SIGNAL-TO-NOISE METHODS

We investigate the use of dual aggregate SDs. Assuming that scores produced by an engine consist of two components, signal and noise, the score random variable S can be decomposed as:

$$S = S_{\text{SIGNAL}} + S_{\text{NOISE}}$$

The probability densities of the components are given respectively by p_{SIGNAL} and p_{NOISE} defined across the engine's output score range.

Furthermore, we assume 'stable' system characteristics for the engine in the sense that the signal and noise levels at a score depend only on the score. We can define a function which normalizes input scores s into the fraction of the signal at s:

$$\text{S/N:} \quad s' = \frac{p_{\text{SIGNAL}}(s)}{p_{\text{SIGNAL}}(s) + p_{\text{NOISE}}(s)} \tag{1}$$

Since engines are expected to produce increasing signal-to-noise ratios as score increases, this may be an interesting normalization.

However, the magnitude of the original score is not taken into account. An obvious improvement would be to multiply S/N with a calibrated score s, for which we could use the HIS normalization:

$$\text{S/N}*\text{HIS:} \quad s' = \frac{p_{\text{SIGNAL}}(s)}{p_{\text{SIGNAL}}(s) + p_{\text{NOISE}}(s)} P(S_{\text{HIS}} \leq s) \tag{2}$$

[1]We simplify their proposal by removing the quantile function that only gives a constant transformation which doesn't impact DIR.

The resulting scores would be comparable across engines, however, the distribution of the variable S_{HIS} depends on the availability of historical queries. Using historical queries, although very feasible and no cooperation is required, may lead to instabilities and biases. To deal with this, we can instead use the variable S_{SIGNAL}:

$$\text{S/N}*\text{SIG}: \quad s' = \frac{p_{\text{SIGNAL}}(s)}{p_{\text{SIGNAL}}(s) + p_{\text{NOISE}}(s)} P(S_{\text{SIGNAL}} \leq s) \qquad (3)$$

This calibrates s to the probability of having signal at or below s.

The question is how to approximate p_{SIGNAL} and p_{NOISE} per engine. Seeing engines as black-boxes similarly to the historical CDF approach, we can feed each one with queries of appropriate types and generate the needed functions based on the statistical properties of the observed output scores.

4. QUERY MODELS

We develop two models for generating artificial queries given a document collection. The resulting query sets produce aggregate SDs approximating S_{NOISE} (monkey query model) and S_{SIGNAL} (human query model).

Monkeys on Modified Typewriters In parallel to the popular thought experiment of a monkey hitting keys at random on a typewriter, let us imagine a keyboard with the terms of a query language on its keys plus "enter". The keys are considered equally accessible and of equal size, except "enter" which has a different size and thus different probability to be hit if keys are hit at random. The monkey, not understanding the grammar and semantics of the query language, will select terms uniformly. Moreover, terms will be independent. If p is the probability of hitting "enter", then the probability that the monkey will type k terms before hitting "enter" is given by (the discrete analogue of the *exponential distribution* called) the *geometric distribution*:

$$P(K_1 = k) = (1-p)^k p, \quad k = 0, 1, 2, \ldots$$

Note that a p fraction of the total queries will be of zero-length. The mean query length will be $1/p$.

Assuming r monkeys using identical keyboards (characterized by the same p) are typing independently, the random variable $K = \sum_{m=1}^{r} K_m$, where K_m is the geometrically distributed variable associated with the mth monkey, follows a *negative binomial distribution*:

$$g(k; r, p) = \binom{k+r-1}{k} p^r (1-p)^k, \quad k = 0, 1, 2, \ldots$$

Under an alternative parameterization, $\lim_{r\to\infty} g(k; r, p)$ converges to the *Poisson distribution* with a rate $\lambda = r(1/p - 1)$:

$$\text{Poisson}(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Humans on Search Engines Query terms occur, in general, in a dependent way (i.e. the occurrence of one makes the chances of occurrence of some others better than random) due to all of them pointing at the same topic. For natural language queries, there exists also serial dependence, imposed by grammar and semantics. When incorporating dependencies, retrieval models are becoming practically intractable, which led in the past to the infamous *term independence assumption*. Instead of trying to model term probabilities of occurrence and dependencies, we can rather tackle both features at once by picking real text fragments out of a corpus. The remaining question is how long those fragments should be.

Arampatzis and Kamps [1] arrive at a truncated Poisson/Power-law model of query length. The bulk of queries concentrates at short lengths where a power-law does not fit at all given the current query languages, therefore it makes practical sense to use a truncated mix of Poisson-Zipf to generate query lengths. In such a practical model, the lengths are Poisson-distributed for $k < k_0$ while they are Zipf-distributed for $k \geq k_0$. The choice of k_0 depends on the specific domain (i.e., a combination of features of the document collection, query/indexing language, and pattern of use of the system). As a rule of thumb, k_0 seems to be just above the mean observed query length.

Table 1: Distributed retrieval results for TREC-123 and TREC-4 over all 100 engines. Significant-tested with a bootstrap test, one-tailed, at significance levels 0.05 (°), 0.01 (°), 0.001 (•).

	TREC-123			TREC-4		
run	P10	P20	P30	P10	P20	P30
ROUNDROBIN	0.1835	0.1835	0.1835	0.0584	0.0584	0.0584
Z-SCORE	0.2320°	0.2285°	0.2167°	0.1300•	0.1130•	0.0940°
HIS	0.2340°	0.2120 -	0.2017 -	0.1920•	0.1540•	0.1487•

Table 2: Distributed retrieval results for TREC-123 and TREC-4 over all 100 engines.

	TREC-123			TREC-4		
run	P10	P20	P30	P10	P20	P30
HIS	0.2400	0.2165	0.2047	0.1920	0.1540	0.1487
S/N	0.2630 -	0.2495°	0.2290 -	0.1980 -	0.1740 -	0.1560 -
S/N∗HIS	0.3020•	0.2770•	0.2537•	0.2380°	0.1920°	0.1740°
S/N∗SIG	0.3380•	0.3095•	0.2790•	0.2400°	0.2090•	0.1793°

5. EVALUATION: DIR TESTBEDS

Standard score normalization methods like the MinMax ignore the score distribution: $s' = \frac{s-min}{max-min}$, with min (max) the minimal (maximal) score per query and engine. That is, MinMax forces all scores in [0,1], resulting in a maximal score per topic and engine of 1. In DIR, we will be doing effectively a ROUNDROBIN picking the top result of each engine. We calculate also the Z-SCORE over the top 1,000 results, which is much more effective than ROUNDROBIN (see Table 1). The historical CDF approach HIS is also significantly better than ROUNDROBIN, and at least as good as Z-SCORE. We compare HIS against the new signal-to-noise methods S/N, S/N∗HIS, and S/N∗SIG. Table 2 presents the distributed retrieval results without resource selection. Overall, the S/N∗HIS and S/N∗SIG runs show significant improvements over the strong baseline of HIS, while the consistent improvements in S/N are mostly non-significant.

Acknowledgments This research was supported by the Netherlands Organization for Scientific Research (NWO), CATCH programme, under project number 640.001.501.

REFERENCES

[1] A. Arampatzis and J. Kamps. A study of query length. In *Proceedings SIGIR'08*, pages 811–812. ACM, 2008.

[2] A. Arampatzis and J. Kamps. A signal-to-noise approach to score normalization. In *Proceedings CIKM 2009*, pages 797–806. ACM Press, New York USA, 2009.

[3] M. Fernández, D. Vallet, and P. Castells. Using historical data to enhance rank aggregation. In *Proceedings SIGIR'06*, pages 643–644. ACM, 2006.

[4] S. Robertson. On score distributions and relevance. In *Proceedings of 29th European Conference on IR Research (ECIR'07)*, pages 40–51. Springer, 2007.

Design and Evaluation of a University-wide Expert Search Engine

Extended Abstract

Toine Bogers
Information Interaction and Architecture
Royal School of Library and Information Science
Birketinget 6, DK-2300
Copenhagen S, Denmark
tb@db.dk

Ruud Liebregts
Textkernel BV
Nieuwendammerkade 28A-17
NL-1022 AB
Amsterdam, The Netherlands
liebregts@textkernel.nl

ABSTRACT

The ability to discover individuals that are knowledgeable about a certain topic, task, or assignment is essential for organizational effectiveness and is generally referred to as expert search. So far, most of the existing expert search engines have been evaluated and compared under laboratory conditions using static collections, whereas large-scale user evaluations of expert finding systems have been largely absent. In addition, little objective evidence has been presented on the benefits of such dedicated expert search engines or their ability to outperform the existing resources organizations have in place on this task.

In this paper we report on the design and large-scale evaluation of an expert search engine for Tilburg University (UvT), a medium-sized university in the Netherlands. The search engine taps into a variety of bilingual sources of topical expertise evidence for over 1,900 university researchers. Tilburg University currently has four different information sources available that can be used to support manual expert finding. Two of these sources are document repositories that contain over 40,000 scientific publications and over 12,500 student theses. Users can search through the metadata fields for relevant publications or theses, but full text search is not available. A third information source is Webwijs ("Webwise"), an online database of university experts and expertise where researchers maintain their own profile. Researchers can enter research descriptions and select expertise areas from a predefined list. It also links to courses taught by the expert and authored publications. Users can search for experts on Webwijs by expert name—unknown in an expert finding scenario—or by topic. Searching by topic has the disadvantage that the Webwijs system only accepts queries that match one of the predefined expertise key words, reducing flexibility. Finally, the fourth source of information is the search engine for the UvT intranet. These retrieved Web pages are the most difficult of all three sources to associate with candidate experts.

With so many fragmented sources of expertise evidence as in the UvT situation, it is difficult for users to obtain a coherent picture. Our expert search engine therefore combines the content-based evidence available on Webwijs and from the repositories. To this end, we extracted all English and Dutch documents from the publication and thesis repositories, indexing all the relevant metadata, such as title, author(s), and publication date, and the full text when available. People were unambiguously associated with publications using the unique univerity IDs (ANR) of the authors. ANRs of thesis supervisors were not available in the repository, so we used pattern matching techniques to match the metadata names to candidate experts, resulting in a small percentage of possible association errors. We additionally crawled the Webwijs page of each candidate expert, if available. All information from Webwijs—research descriptions, expertise areas, and course descriptions—were stored in a single Webwijs profile document for each candidate expert. The generated XML documents were indexed using the Indri toolkit. English and Dutch stop words were removed and English Krovetz stemming was applied to all documents. In this approach, we built on previous work by basing our data on an updated and expanded version of the UvT Expert Collection.

In our search engine, we opted for a document-centric approach to expert finding. Previous work has shown that a document-centric approach to expert finding works well and that is a robust model for expert finding. A document-centric approach consists of three steps: (1) normal document retrieval to match the query with relevant documents in the collection, (2) associating the retrieved documents are associated with the candidate experts, and (3) expertise attribution, where a relevance score is assigned to each candidate expert and the top relevant experts are presented to the user. For document retrieval, we used the Indri toolkit and selected Dirichlet smoothing as our retrieval model. Expert association was performed at indexing time. Expertise attribution can be done in different ways. Preliminary experiments suggested using a weighted combination of the rank reciprocal of a document and the document's relevance score on that query.

DIR 2010 January 25, 2010, Nijmegen, the Netherlands.

We evaluated our expert search engine in three different ways: (1) a system-based evaluation, (2) an expert-based evaluation, and (3) a user-based evaluation. The goal of our system-based evaluation was to evaluate the different options available during the design stage in a laboratory setting to determine the optimal settings for our expert search engine. For this evaluation, we needed a set of representative queries and the appropriate relevance judgments at the expert level. We constructed this using the known associations between documents and experts: we assumed that the document authors or supervisors were the relevant experts for topical queries derived from those documents. We randomly selected 120 publications and 120 theses and divided these equally over the English and Dutch documents in our collection. For each of these random documents, a short query topic was derived from the title and abstract of each document, with average query length being 2.1 words. This resulted in a set of 240 queries, 120 for each language, each subset spanning 60 documents and 60 theses. Relevance judgments were assumed to be binary and the relevant experts for each query were the associated authors or supervisors. The documents from which the queries were derived, were removed from the result lists for those queries. Optimization using this test collection suggested using language modeling with Dirichlet smoothing and Indri's dependence model, as well as pseudo-relevance feedback.

In our expert-based evaluation, we enlisted the help of 30 UvT researchers, selected proportionately from the different faculties, to create a new test set with realistic topics and reliable relevance judgments. Before exposing participants to the expert search engine, we asked each of them to write down one topic about which they themselves were knowledgeable. We also asked them to name up to five other experts on that topic, also employed by the UvT. Participants were asked to rate their own and their colleagues' relative expertise level on a five-point scale. Participants were then shown the search engine and asked to use it to find experts on their pre-specified topic. Participants were asked to judge each of the top 10 candidates returned by the search engine. We concluded with a short survey to measure user satisfaction.

This resulted in an 'expert-centered' set of 30 new queries with 268 graded and realistic relevance judgments, enabling us to verify our findings from the system-based evaluation. Re-testing our original design options using this new query set did not result in having to change the expert search engine's retrieval parameters. Finally, the survey results showed a high degree of satisfaction with the search results.

In our third and final evaluation step we involved real users to benchmark the performance of the expert search engine ('new system') against all other information sources currently available within Tilburg University combined ('old system'). We used two different groups of participants: an internal group of UvT students who had prior knowledge of the current systems and may have been familiar with some of the topics and experts, and an external group consisting of high school students who had no such prior knowledge.

In this experiment, users were given two different types of simulated work tasks: expert finding and thesis supervisor finding. Each work task included a description of the topic and an indicative request that expressed the task type. We used the topic set from the expert-based evaluation to construct five work tasks of each type and used the associated relevance judgments to evaluate the selections participants made. In addition, we performed a manual relevance assessment of the candidate experts that were recommended frequently by our participants, but were not covered by the expert-based relevance judgements. In order to reduce the variation in search strategies used by participants, we used a within-subjects design, in which all participants used both system types an equal number of times (if possible). Display order of the tasks was random, and task types and systems to be used were also assigned randomly. All participants were required to complete at least four tasks and at most ten, and they were asked to select up to three relevant experts.

The experiment was entirely Web-based so users could participate from any location. The tasks and the expert selections were shown in a pane on the left side of the screen, as well as the available sources for each current task. Selecting a source—the new system or any of the old systems—opened the search engine interfaces in a large frame to the right of the task pane, so that both were visible simultaneously. At the end of the experiment, we conducted a short survey with questions that together covered all aspects of usability: effectiveness, efficiency, and satisfaction. To benchmark system performance, we used the immediate accuracy and qualified search speed metrics. An advantage of these measures is that they are proportional and can therefore be used in cross-system comparisons.

A total of 101 users participated in the experiment, of which 44 were part of the external group and 57 were internal participants, selected equally from all faculties. A total number of 325 tasks were completed using the new expert search engine and 332 using the old system. Our expert search engine was found to be significantly more effective than the old system, as measured by immediate accuracy. In addition, we found that users with no prior experience with the old system are at a disadvantage. In fact, the external participants even performed slightly better than the internal group when using the expert search engine, which suggests there is almost no learning curve for the expert search engine. Evaluation using qualified search speed showed that the expert search engine is both more effective and efficient. For the expert search engine, the majority of the relevant answers were in high relevance categories. The expert search system also had more relevant answers in the highest category than the old system, while the old system had more irrelevant answers. The external group benefits most from using the expert search engine in terms of search speed, once again illustrating the absence of a learning curve of the new system. Finally, users also showed high satisfaction with all aspects of the search engine.

Our results show that an integrated approach to expert finding, such as our expert search engine offers, yields several benefits. Our search engine scores significantly better on user satisfaction, efficiency, and effectiveness when benchmarked against the current systems available at the university. Users find more highly relevant answers using our search engine, and find them significantly faster. Perhaps most importantly, we showed that our integrated approach has no learning curve for outside users.

A Machine Learning Approach to Sentiment Analysis in Multilingual Web Texts

[Extended Abstract] *

Erik Boiy
Department of Computer Science, K.U.Leuven
Leuven, Belgium
erik.boiy@cs.kuleuven.be

Marie-Francine Moens
Department of Computer Science, K.U.Leuven
Leuven, Belgium
sien.moens@cs.kuleuven.be

ABSTRACT
Sentiment analysis, also called opinion mining, is a form of information extraction from text of growing research and commercial interest. In this paper we present some findings of our machine learning experiments with regard to sentiment analysis in blog, review and forum texts found on the World Wide Web and written in English, Dutch and French.

Categories and Subject Descriptors
I.5.2 [**Pattern Recognition**]: Design Methodology

General Terms
Languages, Design, Experimentation

Keywords
Opinion mining, information tracking, cross-language learning, active learning

1. INTRODUCTION
For this research we are interested in the feelings that people express with regard to certain consumption products. We learn several classification models from a set of examples that are manually annotated as being positive, negative or neutral with regard to a certain entity of interest. We define an entity as the non-abstract subject matter of a conversation or discussion, e.g., a movie or a new car model, towards which the writer can express his or her views. We have to deal with several problems, such as the noisy character of the input texts, the attribution of the sentiment to a particular entity, and the small size of the training set. In addition, we study problems of the portability of the learned models across domains and languages. Finally, we compare several active learning methods to be able to reduce the amount of examples that need to be manually labeled.

*Full version published in Information Retrieval, available at: `www.springerlink.com/content/p01j644041467237`

2. METHODOLOGY
We build our models using well-known features such as unigrams and stems, augmented with linguistic features such as negation (indicating a reversal of the affected features' polarity), comparisons (having a positive and a negative end), discourse features (stressing the importance of certain features) and special handling of compound words and composite verbs (for the Dutch language). We use parse tree information to determine the importance of features in relation to the entity of interest. Either the difference in depth between the word feature and the entity in the parse tree determines the feature weight, or the parse tree is seen as a graph and the length of the path between the feature and the entity (using a breadth-first search) is used for assigning a weight to the feature.

We performed our experiments using a common selection of machine learning techniques: Multinomial Naïve Bayes (MNB), Support Vector Machines (SVM) and Maximum Entropy (ME) models. The classification models can be configured in a cascaded pipeline, with certain conditions guarding against going deeper into the cascade. The deeper in the cascade, the more expensive the features become. We experimented with several setups before arriving at a three layer cascade. In the first layer we tackle the problem of having an abundance of neutral examples, of which a large portion are uninformative or advertisement messages. This layer consists either of one classifier or of a bagged aggregation of several classifiers, the latter with the purpose of obtaining multiple proofs that an example is actually neutral. In the second layer of the cascade, our three-class classification into positive, negative and neutral sentiment is performed. In the third step of the layer, classification of difficult cases is based on the extraction of expensive features obtained by parsing the sentence into its syntactic dependency structure.

Lastly, we used and combined three active learning methods from the literature. Using Uncertainty Sampling, the examples for which the current classification model is most uncertain are selected for labeling. Relevance Sampling does the opposite: the examples which are most likely to be class members are selected. This technique is used to find more examples for the minority class "negative". In order to achieve diversity among chosen examples, Kernel Farthest First selects examples that are far away (according to a chosen kernel function) from the examples currently labeled.

Table 1: Our best results in terms of accuracy, precision, recall and F-measure (F_1) using the English (a), Dutch (b) and French (c) corpora and the setups indicated in Section 4. For English, Dutch and French we implemented respectively an MNB, an SVM and an ME classifier – 10 fold cross-validation.

(a) English

Architecture	**Accuracy**	**Precision** pos/neg/neu	**Recall** pos/neg/neu	**F-measure** pos/neg/neu
Cascade	83.30	69.09/85.48/85.93	55.73/82.40/91.84	61.70/83.91/88.79
SC uni-lang	83.03	69.59/86.77/85.08	56.13/79.60/92.12	62.14/83.03/88.46
SC uni	82.73	68.01/85.63/85.53	58.40/78.67/91.24	62.84/82.00/88.29

(b) Dutch

Architecture	**Accuracy**	**Precision** pos/neg/neu	**Recall** pos/neg/neu	**F-measure** pos/neg/neu
Cascade	69.03	63.51/53.30/72.20	42.93/31.20/88.20	51.23/39.36/79.40
SC uni-lang	69.05	60.39/52.59/73.63	49.60/33.87/85.44	54.47/41.20/79.10
SC uni	68.18	58.73/49.58/73.24	48.00/31.73/85.16	52.82/38.70/78.75

(c) French

Architecture	**Accuracy**	**Precision** pos/neg/neu	**Recall** pos/neg/neu	**F-measure** pos/neg/neu
Cascade	67.68	50.74/55.88/71.90	27.47/38.67/88.44	35.64/45.71/79.32
SC uni-lang	65.97	47.67/50.33/72.18	30.00/40.67/84.36	36.82/44.99/77.79
SC uni	65.83	45.67/50.82/72.23	28.8/41.33/84.28	35.32/45.59/77.79

3. CORPORA

We collected three corpora – respectively composed of English, Dutch and French sentences – from blog, review and news forum sites. There is no clear border between the types of text they represent, e.g., all sources may contain reader comments. Sources include major blog sites, e.g., skyrock.com, livejournal.com, xanga.com, blogspot.com; review sites, e.g., amazon.fr, ciao.fr, kieskeurig.nl; and news forum sites, e.g., fok.nl, forums.automotive.com. The sources reflect realistic settings for sentiment extraction and contain a mixture of clean and noisy texts. As entities we considered car brands and movie titles. These corpora were handlabeled and are proprietary.

4. SELECTED RESULTS

Table 1 gives the general performances for each language. We compare a cascaded approach with two single classifiers (*SC*). The first (*SC uni-lang*) uses unigram, discourse and negation features, as described in Section 2. The second (*SC uni*) uses only unigram features. In the cascaded approach (*Cascade*) unigram features are used in the first layer, unigram, discourse and negation features are used in the second and third layers. In the third layer parse features are added. The first and second layer of the cascade and the single classifiers were trained with nine tenth of the corpus. The third layer of the cascade was trained with all parsable positive and negative examples that were classified correctly in an isolated cross-validation of the second layer. When testing the cascade, examples with a certainty above a given threshold move to layer three. Examples that cannot be parsed in the third layer of the cascade keep the classification given by the second layer.

For the Dutch corpus, layer three has an overall negative effect on the results. When using only layers one and two however, results improve to achieve an accuracy of 69.8%.

5. CONCLUSIONS

It was found that unigram features augmented with a limited number of language-specific features yield accuracy results of ca. 83%, 70% and 68% when classifying English, Dutch and French Web data, respectively, and slightly improve a baseline classification which only uses unigrams. A cascaded approach that reserves the computation of expensive features to a subset of the sentences further down in the cascade could only slightly positively influence accuracy and F-measures, but allowed to test a number of hypotheses. Among them, we found that the performance is increased by first filtering neutral sentences. In the literature this filtering was found useful when classifying complete review documents, and our tests confirm this finding for classifying individual sentences. Incorporating a layer in the cascade where expensive parse features are used, improved the performance for classifying sentences in which sentiments are expressed towards different entities.

Sparsity of training examples was an important cause of errors, which is especially severe in case the language of the sentences diverges largely from formal language (as is the case for French blogs). Active learning, especially when combining several methods, provides a small, but noticeable improvement in average F-measures over randomly selecting examples for labeling, making it possible to arrive at better results when labeling an equal amount of examples. As the sentiment analysis might be extended to include new domains and different languages, even a small benefit here is very valuable.

6. ACKNOWLEDGMENTS

We are grateful to the IWOIB (Institute for the Encouragement of Scientific Research and Innovation of Brussels) and the company Attentio, Belgium, who sponsored this research, and to all others that made a contribution.

An Overview of Approaches to Extract Information from Natural Language Corpora

Frederik Hogenboom
fhogenboom@ese.eur.nl

Flavius Frasincar
frasincar@ese.eur.nl

Uzay Kaymak
u.kaymak@ieee.org

Econometric Institute
Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands

ABSTRACT

It becomes increasingly important to be able to handle large amounts of data more efficiently, as anyone could need or generate a lot of information at any given time. However, distinguishing between relevant and non-relevant information quickly, as well as responding to newly obtained data of interest adequately, remain cumbersome tasks. Therefore, a lot of research aiming to alleviate and support the increasing need of information by means of Natural Language Processing (NLP) has been conducted during the last decades. This paper reviews the state-of-the-art of approaches on information extraction from text. A distinction is made between statistic-based approaches, pattern-based approaches, and hybrid approaches to NLP. It is concluded that it depends on the user's need which method suits best, as each approach to natural language processing has its own advantages and disadvantages.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*knowledge acquisition*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*text analysis*; I.5.4 [**Pattern Recognition**]: Applications—*text processing*

General Terms

Languages, algorithms

Keywords

Information extraction, natural language processing (NLP), text mining, parsing

1. INTRODUCTION

In today's busy, data-driven world where the stream of vital information never ends, anyone could generate or collect a lot of information about almost anything at any given time. Due to the enormous and ever growing amount of information that is available, it becomes increasingly important to be able to handle these large amounts of data more efficiently. For instance, one has to be able to distinguish between relevant and non-relevant information quickly and respond to newly obtained data of interest adequately.

As a consequence, a lot of research aiming to support the increasing need of information has been conducted during the last decades. At first, research was mainly focused on Information Retrieval (IR), but currently the focus has shifted to Information Extraction (IE) or data mining. An omnipresent problem is the fact that most data is unstructured, being described using natural (human-understandable) language. In order to retrieve or process large amounts of data, it is desired to make use of machines. However, natural languages are not machine-understandable and thus there is a need for performing Natural Language Processing (NLP) [2].

NLP is a field in computer science and linguistics that is closely related to Artificial Intelligence (AI) and Computational Linguistics (CL). NLP is generally employed to convert information stored in natural language to a machine-understandable format. Thus, the main goal of NLP and IE is to extract knowledge from unstructured data. The main difficulties that are encountered with NLP arise when longer sentences that are highly ambiguous and have complex grammars are to be processed. The challenges imposed by automatically processing natural language have motivated the ongoing research into NLP for several decades.

Throughout the years, many NLP systems have been created, and nowadays, innovative NLP systems are still being developed, as the popularity of NLP witnesses a substantial growth, caused by, for example, the huge amount of available (electronic) text and the presence of adequate processing power. NLP systems vary in employed techniques, are built for different purposes, and may differ in focus. In general, one can distinguish between several layers within the processing tasks performed by an NLP system [1]. These levels of language range from the phonology and morphology of elements, to the lexical, syntactic, and semantic aspects of text, to the discourse and pragmatic properties of natural language text. Some systems focus more on the lower levels of processing, whereas other systems focus on the higher levels or on all levels. Generally speaking, three main approaches to NLP exist, i.e., statistics-based, pattern-based, and hybrid approaches.

DIR 2010 January 25, 2010, Nijmegen, the Netherlands.

2. STATISTICS-BASED APPROACHES

Statistical approaches are commonly used for natural language processing applications. These methods are data-driven and rely solely on (automated) quantitative methods to discover relations. Statistical approaches require large text corpora in order to develop models that approximate linguistic phenomena. Furthermore, statistics-based NLP is not restricted to basic statistical reasoning based on probability theory, but encompasses all quantitative approaches to automated language processing, such as probabilistic modeling, information theory, and linear algebra.

Even though one could distinguish between word-based and grammar-based approaches (e.g., word frequency counting and part-of-speech tagging, respectively), all statistics-based approaches to NLP share their focus on discovering statistical relations, i.e., facts that are supported by statistical evidence. Examples of discovered facts are words or concepts that are (statistically) associated with one another. It should be noted that statistical relations do not necessarily imply semantically valid relations or relations that have proper semantic meaning.

Hence, statistical methods do not deal with meaning explicitly, i.e., they discover relations in corpora without considering semantics. However, from a statistical point of view, this is a matter of definition more than it is a real issue. One could argue that true meaning is not related to philosophical semantics, but to evidence that resides within the distribution of contexts over which words and utterances are used. Another disadvantage of statistics-based NLP is that it requires a large amount of data in order to result in statistically significant results. However, these approaches are not based on knowledge, and thus neither linguistic resources, nor expert knowledge are required.

3. PATTERN-BASED APPROACHES

In contrast to statistics-based approaches, pattern-based approaches are based on linguistic or lexicographic knowledge, as well as existing human knowledge regarding the contents of the text that is to be processed. This knowledge is mined from corpora by using predefined or discovered patterns. One could distinguish different types of patterns, i.e., lexico-syntactic and lexico-semantic patterns. The former patterns combine lexical representations and syntactical information with regular expressions, whereas the latter patterns also employ semantic information. These semantics are added by means of gazetteers (which use the linguistic meaning of text) or ontologies (which also include relationships).

There are several advantages that result from the utilization of pattern-based approaches to perform NLP tasks over statistics-based approaches. First of all, pattern-based approaches need less training data than statistical NLP approaches. Also, it is possible to define powerful expressions by using lexical, syntactical, and semantic elements, and results are easily interpretable. Patterns are useful when one needs to extract very specific information. However, in order to be able to define patterns that retrieve the correct, desired information, lexical knowledge and possibly also prior domain knowledge is required. Other disadvantages are related to defining and maintaining patterns, as these are cumbersome and non-trivial tasks.

4. HYBRID APPROACHES

Although theoretically there is a crisp distinction between statistical and pattern-based approaches, in reality, it appears to be difficult to stay within the boundaries of a single approach. Often, an approach to NLP can be considered as mainly statistical or pattern-based, but there is also an increasing number of researchers that equally combine data-driven and knowledge-driven approaches, to which we refer to as hybrid approaches. For instance, it is hard to apply solely pattern-based algorithms successfully, as these algorithms often need for instance bootstrapping or initial clustering, which can be done by means of statistics. Furthermore, hybrid approaches to NLP could emerge when solving the lack of expert knowledge for pattern-based approaches, by applying statistical methods. Also, researchers can combine statistical approaches with (lexical) knowledge, for instance to prevent unwanted results.

By combining different techniques, advantages as well as disadvantages of statistical and pattern-based approaches are inherited. For instance, one is able to create complex patterns and less data is required compared to statistical approaches, but more data is required compared to pattern-based methods. Some inherited disadvantages can be (partially) cleared by advantages, e.g., the lack of semantics in statistical methods is solved when adding patterns. Disadvantages of hybrid approaches to NLP are related to the multidisciplinary aspects of hybrid NLP systems.

5. CONCLUSION

In this survey, we have elaborated on the main approaches to natural language processing. Each of the approaches has its advantages and disadvantages, and thus we can define guidelines regarding the selection of a proper NLP approach. If one is less concerned with semantics and assumes that knowledge lies within statistical facts on a specific corpus, it is advised to use statistics-based approach. Otherwise, if one is concerned with the semantics of discovered information, or it is desired to be able to easily explain and control the results, a pattern-based approach is more suitable. However, if one needs to bootstrap a pattern-based approach using statistics (for instance when there is insufficient expert knowledge available) or the other way around (e.g., when there is a need for a priori knowledge), a hybrid approach is more appropriate.

6. REFERENCES

[1] E. D. Liddy. *Encyclopedia of Library and Information Science*, chapter Natural Language Processing, pages 2126–2136. Marcel Decker, Inc., 2nd edition, 2003.

[2] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1st edition, 1999.

Measuring children's search behaviour on a large scale

Emiel Hollander, Theo Huibers, Hanna Jochmann-Mannak, Paul van der Vet

Human Media Interaction,
University of Twente
PO Box 217, 7500 AE Enschede

{e.s.hollander,t.w.c.huibers,h.e.mannak,p.e.vandervet}@ewi.utwente.nl

ABSTRACT

Children often experience problems during information-seeking using traditional search interfaces and search technologies, that are designed for adults. This is because children engage with the world in fundamentally different ways than adults. To design search technologies that support children in effective and enjoyable information-seeking, more research is needed to examine children's specific skills and needs concerning information-seeking. Therefore, we developed an application that can monitor children's search behaviour on a large scale. In this paper, we present the steps taken to develop this application. The basis of the application is UsaProxy, an existing system that is used to monitor the user's usage of websites. We have increased the accuracy of UsaProxy and have developed an application that is able to extract useful information from UsaProxy's log files.

Categories and Subject Descriptors

H1.2 [**Models and Principles**]: User/Machine Systems – *Human factors, Human information processing.* H.3.3 [**Information Storage and retrieval**]: Information Search and Retrieval – *Query formulation, Search process, Selection process*

General Terms

Measurement, Documentation, Human Factors.

Keywords

Children's search behaviour, data logging.

1. INTRODUCTION

Interactive technology plays an important part in children's lives. Every day, more children have access to the internet and more information becomes available for them through the internet. The question is if existing search technologies support children in effective and/or enjoyable information-seeking.

Children's search behaviour has not had a lot of attention in research over the past few years. It is quite interesting, however, because children's search behaviour differs from the behaviour of adults in many ways. It also differs between various age groups. For example, young children who have just learned to read, may search using only one or a few words, may make more errors when typing and may benefit from images while browsing the results. Therefore, examining children's search behaviour to design search technologies that support children in effective and enjoyable information-seeking, is an important research topic. We will discuss this in more depth in the next section.

Currently, the search behaviour of children is mostly examined in an experimental setting using additional equipment like eye-tracking devices to observe and record search behaviour [10]. This produces a large amount of useful high-quality data, but children and their parents need to travel to a place where the experiment can be held and the method is very time consuming. The number of children that can participate in such experiments is thus limited.

We were interested in finding a way to make it easier for children to participate in research on search behaviour and to gather data from a far more larger group of children. Therefore, we developed an application that can monitor children's search behaviour on a large scale. This application may be installed on any computer, and may also be accessed through the internet. Using a network of libraries, primary schools and interested parents, we will be able to measure children's search behaviour all over The Netherlands. In our research, we are mostly interested in children from eight through twelve years old.

To our knowledge, there are no studies conducted that use log analysis techniques to examine children's search behaviour on a large scale. Measuring search behaviour on a large scale, using deep log analysis techniques, is far more common with research on adults' search behaviour. These studies, however, are mostly aimed at evaluating the usability of specific websites or applications and not aimed at examining what principles are underlying on the search behaviour of the adult users. For example, Nicholas et al. [16] evaluate the usability of digital scholarly journals using deep log analysis techniques.

Although there are limitations to what one can measure with log analysis as compared to experiments in a controlled environment, this is offset by the fact that a far larger group of children can be involved. We believe that quantitative data from this type of research can provide interesting hypotheses that can be examined in more depth in experimental settings. This makes the research into the application useful.

This report follows the steps that were needed to develop the application to monitor children's search behaviour. First, we determined which variables are useful to measure in assessing children's search behaviour (Section 4). After that, we compared the variables that we wanted to measure with existing applications or systems, to see which one matched our wishes best (Section 5 and 6). Even the best match did not offer the optimal solution for our research goals. Therefore, we needed to adapt the best matching system to our situation. Finally, the characteristics of the application that we developed to assess the usage of information retrieval systems by children, will be discussed in Section 6.

The 10th Dutch-Belgian Information Retrieval Workshop (DIR 2010), January 25, 2010, Nijmegen, The Netherlands.

Before discussing the development of the tool in more depth, we will first discuss how children are different from adults and what problems these differences cause for children in using digital technologies. By discussing these problems, we want to stress the importance of developing search technology that is tailored to children's specific needs and skills. We conclude that more research is needed to examine children's search behaviour and the developed tool presented in this paper can be used for this type of research on a large scale.

2. PREVIOUS RESEARCH ON CHILDREN'S SEARCH BEHAVIOUR

Design principles that are applied on search interfaces for children are often a projection of adult's vision about children's preferences. Interface designers take adult media content and attempt to make it 'childlike', by simplifying the content and adding more visual design (e.g. lots of colours) and multimedia (e.g. videos and animations) [17]. Also, children are often thought to be as web savvy as adults and sometimes they are even thought to understand technical terminology better than their parents. Are children that web savvy as most people think? Do they find information that easily using search technologies that are initially designed for adults?

2.1 Why do children experience problems?

Developmental psychologists show why children are fundamentally different from adults. Children are not just little adults that only lack knowledge and experience in comparison to adults. They fundamentally experience and understand the world different than adults [9]. Jean Piaget [18], for example, described in his earlier works how children's cognitions evolved during a series of four stages from sensorimotor (birth to 2 years), to preoperational (ages 2 to 7), to concrete operational (ages 7 to 11) to finally, the formal operational stage of development (ages 11 and up). Contemporary research recognizes that all children develop differently, but that Piaget's general characterizations of children still remain useful. Also when examining children's interactions with digital technology.

Because of these fundamental differences between children and adults, children can experience all kind of problems while exploring digital environments and using digital technologies.

2.2 Dexterity

Children's motor skills are not equal to that of adults. Therefore, traditional input devices can be difficult for children to use. They have difficulties holding down the mouse button for extended periods or to drag-and-drop objects using the mouse [12]. Also typing can be difficult for children, because they have to 'hunt and peck' on the keyboard for the correct keys [7].

2.3 Problems with searching

There are several reasons why children have more difficulties with formulating search queries than adults. Children have less knowledge to base 'recall' on than adults [7, 10] and they rarely access their previous knowledge of the topic during formulating search queries [21]. They also do not have a very developed vocabulary as adults do and they have difficulties with correct spelling, spacing and punctuation, which is needed for most search engines to find relevant search results. Also, moving from natural language to a single keyword is more difficult for children than for adults [20].

2.4 Browsing problems

Browsing can also be more difficult for children than for adults. Children have difficulties to understand and select abstract terms; search tasks are more successful when concrete terms are used [7] and children find it easier to retrieve concrete terms than abstract terms [4]. Children can also have trouble understanding categories and finding the right category, because they have less domain knowledge and less vocabulary knowledge than adults [7]. The same problems occur with the use of metaphors from the adult world, such as file folders or filmstrips, which are unfamiliar to most children [9].

2.5 Interaction style

Children's patterns of attention and interaction are quite different from those of adults. Traditional task-oriented analyses of activity does not support the playful, spontaneous nature of children's interaction with technology [9]. Children are more reactive searchers and are more chaotic in their search performance than adults. They make more web moves, backtrack more often, loop searches and deviate more from their designated target [6].

3. RESEARCH OBJECTIVES

This paper reports on research to find a solution for measuring children's search behaviour that can be carried out without the need for a completely tailored experimental setting.

Four questions can be identified that we will need to answer in order to find such a solution. Each of these questions will be discussed in the following sections.

1. Which variables are useful to measure to assess search behaviour of children?
2. Which of these values can be measured using only an application that can be installed or used on any computer?
3. What are the solutions for measuring search behaviour that already exist and which values do they measure?
4. Which of these solutions matches best with what we want to measure, and how does it need to be adapted to make it a perfect match?

4. MEASURABLE VARIABLES

We would like to measure everything that we are able to measure using only an application that can be installed or used on any computer without needing any extra equipment.

4.1 Variable groups

We can divide the variables that can be measured when people are using a search engine in four distinct groups.

1. Variables from measurements done directly on any computer involved in the search process. These are for example the number of clicks, amount of scrolling, speed of typing or which documents are retrieved.
2. Variables from measurements done by external equipment. An example of this are the values obtained by using an eye tracking device.
3. Variables obtained by user input. These are acquired by asking the user questions before, during or after the task.
4. Variables obtained by observation. The researcher sits near the user participating in the experiment and makes notes of everything that happens.

Research on the topic of information retrieval for children covering variables from groups 2, 3 and 4 is available. An example of a method that is aimed at obtaining user information from group 3 from children is The Fun Toolkit [19].

We are, however, in search of an application that can gather information about children's behaviour when using search engines, that can be installed or used on any computer and without the needed interference of a researcher. We therefore can discard everything in group 2 and group 4. Also, since we want to directly observe behaviour, group 3 can be discarded as well.

Group 1 can be further divided into two distinct subgroups.

a. Variables which can only be measured locally on the computer being worked on. Examples are the number of clicks, amount of scrolling and speed of typing.
b. Variables which can also be measured on a server that the computer contacts. These are, for example, the documents that are retrieved or any other kind of server log.

Logging mechanisms on web servers are very common, so data from group 1b is readily available. We are primarily interested in measuring variables from group 1a. This data is not readily available; we need additional applications to gather it.

4.2 Overview of variables

There are several ways to gather a complete list of variables we are able to measure. The first approach is to find out which variables we are technically able to measure, using the input from devices usually attached to a computer. This will deliver variables such as mouse movements, clicks and keystrokes.

The second approach is to define what we would like to know and from there deduce which variables are needed exactly to be able to measure this.

Both approaches will deliver multiple layers of variables. Some that may be measured directly, others that can be deduced from existing measurements.To help perform both approaches, we will first study existing literature to find out which variables other researchers have chosen to measure in similar research.

4.2.1 Existing literature

Bilal has done measurements while children were performing search tasks [5]. She has divided the variables that can be measured into two groups: Transcribed Moves and Selection Actions. These groups are not distinct.

The Transcribed Moves consist of "moves that include all traversal behaviours" and are:

- searching
- browsing
- looping
- backtracking
- screen scrolling
- mouse movements
- exploratory moves

Selection Actions include only

- searching
- browsing (hyperlink activation)
- looping

Of these variables, only screen scrolling and mouse movements belong in group 1a. All the other variables can also be measured on the server that handles the requests (group 1b). Backtracking may not be measurable on the server when the client uses its cache, but this can easily be resolved by disabling the client's cache.

Schacter et al. have also performed analyses while children were performing search tasks [21]. They have logged the following variables:

- time spent on each web page
- time spent on each task
- total number of mouse clicks per task
- keywords entered
- URLs of all web pages visited

From these variables, the mouse clicks clearly belong in group 1a.

Kalsbeek and De Wit have, in cooperation with a primary school teacher, constructed a list of errors commonly made by children when they use search engines [13]. The list also contains a few differences in search behaviour of children when compared to adults. It may be possible to automatically measure some of these errors. A small excerpt from the list that Kalsbeek and De Wit made, is shown below. This list contains the items that can be detected automatically by software.

Table 1: Classes of differences in search behavior between children and adults (excerpt from [13])

Class	Examples
Number words	W8, 4u, xs4all
No vowels	hll wrld (hello world)
Special characters	€pe (Europe)
Smileys	:), ;-)

4.2.2 Defining variables

Using this knowledge, we can now define a complete list of variables that we can measure. Our objective is to make a generic tool for measuring behaviour. The second approach of defining variables is therefore less feasible in this situation. We cannot explicitly define what we want to know, because this will differ for each experiment we will conduct using this tool.

First, we will build a list of variables we can technically measure using the first approach. This is the bottom-most layer of variables; the ones we obtain by directly measuring data. We will call these "device data".

Both Bilal and Schacter have measured mouse movements and clicks. These clearly belong to device data and should therefore be in our list. Both researchers also keep track of which pages are visited. These can directly be obtained from server logs and therefore also belong to this group.

We can directly measure what the user types on the keyboard and what information is on the screen. Below we give an overview of all device data variables.

Level 1: Device data
Keyboard input
Keystrokes
Mouse input
Movement (coordinates)
Clicks (coordinates)
Scrolling

Level 1: Device data
Screen data Screen content
Server logs Pages visited

A timestamp is stored with each piece of device data so that afterwards we can reconstruct exactly what happened at what moment in time. This data can be used directly to deduce a lot of other information. This will be called "directly derived data".

Schacter [21] mentions that he has logged the used keywords. These can be derived from the keystrokes a user has entered. Also logged by Schacter is the time spent on each page, that can also be derived from the server logs.

An overview of all directly derived variables is given below.

Level 2: Directly derived data
From keyboard input Words entered
From mouse input and screen data Buttons and links clicked Items hovered over with cursor
From server logs Amount of time spent on each page

When we have obtained the values for these variables, these can be used to derive even more information. Some examples of information that can be obtained are given below.

Level 3: Indirectly derived data
From words entered Spelling errors
From amount of time spent on page and clicks Path taken through web site
From links clicked and items hovered over Links the user may have hesitated to click on

A variable on level 3 may be an answer to the main research question of an experiment. This question is the starting point for the second approach to obtain all variables: define what we want to know and then deduce which variables are needed. We would then have a research question, find out which variables from level 2 are needed for this and finally, we know which variables from level 1 we need to measure to obtain the information we want.

5. MEASURING VARIABLES

Now that we know which variables we can measure, we need to find an application that is suited best to measure all of these.

Applications that monitor and log user actions do already exist. Broadbent et al. [8] have developed a test case that measures a number of metrics for information retrieval systems. Muresan and Bai [14] have developed a methodology that is aimed at designing the user interface, logger and log analyzer in such a way that as little useful data as possible is lost. More methods like this exist. However, none of them is tailored for usage with children.

There are keyloggers and mouse recorders available for general use, like for example the Keyboard and Mouse Recorder [1]. These tools have, however, not been specifically designed for usage as information source for analyses but rather to simplify computer tasks or for less decent things like stealing passwords.

Also, tools that record exactly what is happening on the screen are widely available. These tools only capture the pixels that are present on the screen and do not log any additional information. This makes analysing these videos a cumbersome task.

Web server logs record which pages have been visited by a person. Unfortunately, that is all they do. They do not register what a user does while on a page, and it is also not always possible to deduct a path the user followed through the site based solely on the web server logs.

WebQuilt tries to solve this problem by having the user visit web pages through a proxy [11]. This proxy records the path a user takes through the site. WebQuilt is able to visualise this information as well; one can easily see which path was chosen most for a certain task. What happens while the user is on a web page is unfortunately not visible when using this tool.

Mueller and Lockerd have developed Cheese, a tool that tracks mouse movement activity on websites [15]. This tool uses embedded scripts to automatically send mouse movement data to the server so that it can be stored there. They have manually visualised this information and evaluated the data they have collected.

Cheese only tracks the position of the mouse on the screen. It does not log what is on the screen and also does not log keypresses. Also, visualisation of what happens is not automatically done by this system.

MouseTrack, another tool specifically designed for websites, does have visualisation options [2]. This tool does not focus on logging mouse clicks, but rather on mouse browsing paths within a website. It is able to display these paths using a variety of visualisation options.

Another tool that is geared towards visualisation of mouse movement on websites is (smt) Simple Mouse Tracking [22]. This tool allows the researcher to exactly replay any user's mouse paths over the original web pages. It is also able to deduce from its logs information about "the user's skills, how he uses the web interface, if he is an impatient person, etc."

These two tools, however, do also not log keyboard input. A tool that does log keyboard input is UsaProxy [3]. The aim of this tool is to log as much as possible while being as unobtrusive as possible to the user. It logs mouse movement, mouse clicks and keyboard input. It also logs which element of the document has been clicked on or has been typed into. Unfortunatly, it does not offer visualisation options.

6. ARCHITECTURES

From the applications listed above, we can extract three architectures that are common for applications like this. These architectures are:

- completely client-side,
- completely server-side,
- using a proxy between the client and server.

We will now further investigate these different architectures.

6.1 Completely client-side

This category consists of applications that only need to be installed on the client and do not need a server to retrieve information from or store information on. The Keyboard and Mouse Recorder [1] is an example of an application that is completely client-side.

The advantage of this kind of applications is that they usually can be installed on any computer. There are no further requirements; install the application and we can immediately start measuring. Another advantage is that we can obtain usage information from all applications running on that computer, and not just for one website or application as would be the case when using a server-side or proxy solution.

The large disadvantage of applications like this is that they can only see where the mouse is on the screen -its actual coordinates- but they cannot see, for example, whether the user is hovering over a link, clicking a button or just clicking randomly.

This is also the case for key loggers. They can see in which application the user is typing, but they cannot see in which field the user is entering information. For our purpose, this is vital information.

There are a number of ways that we could get this additional information using only client-side applications. The first is to also record what is happening on the screen, for example, by using a video recorder. This means that the researcher needs to watch the video to deduce information out of it. When used in combination with a key logger or mouse recorder, we can use the information from these applications to quickly fast forward to potentially interesting moments in the video. This reduces the amount of work needed to analyse the video. It is, however, still not an ideal solution.

The second way is to develop a browser plug-in. Such a plug-in will not have the advantage of being able to capture information about all applications running on the computer. However, it will have complete access to the current web site the user is visiting. It can determine exactly what part of the site the user is interacting with; for example, which link he is hovering over with his mouse, or which input box he is typing text into. We have, unfortunately, not been able to find such a plug-in readily available.

6.2 Completely server-side

Completely server-side may be a misleading title, since we will always need scripts running on the client to collect and send information about, for example, mouse movements and key presses. However, completely server-side implies that we do not need to *install* anything on any client computer that we use for measurements.

Simple Mouse Tracking [22] is an example of a logging application that is completely server-side. The web pages that (smt) is used on, need to have been altered to contain a piece of JavaScript-code which takes care of mouse tracking.

Cheese [15] is another tool that uses this approach. The authors have also embedded scripts in their web pages that collect information about mouse movements.

The advantage of solutions that are completely server-side is that there is no need to install an application on every client that is used to measure variables. Measures will be done for all visitors that visit websites which have these scripts embedded within them.

The disadvantage is that the actual website we want measurements for needs to be altered in order to obtain these measurements. We have to manually embed JavaScript in these websites for these server-side solutions to work.

6.3 Using a proxy

We define a proxy as any entity which fetches the desired website for us and makes sure it is able to carry out measurements on the usage of this website. This entity may be an application, a separate server or something else. One could argue that an application, running on the same server as the application serving web pages, that acts as an extra layer between us and the application serving web pages, is a completely server-side solution. However, using our definition, such an application would be defined as a proxy.

MouseTrack [2] is an example of such an application. It consists of a PHP script that fetches the desired web page and enhances it with JavaScript, that is used to measure usage of the page and also to visualise this data.

UsaProxy [3] uses a different approach. This application is a full-fledged HTTP proxy that will forward any request to the actual web server. It will then modify the responses coming from the web server to contain JavaScript that will collect usage information.

The advantage of this approach is that UsaProxy can be installed, either on the client that is used to visit the web sites, or on the server which delivers the web pages. MouseTrack can only be installed on a computer that runs on a web server.

7. OUR APPLICATION

We have chosen to use UsaProxy for our efforts. This system is the most flexible of all discussed systems, supplies all necessary data and is open source.

UsaProxy embeds JavaScript in all web pages that are requested. This JavaScript is used to send information about events back to the application while the user is browsing sites. An event occurs whenever the user moves the mouse, clicks, types text, etc. To send this information back, UsaProxy makes use of the AJAX functionality that is present in all modern browsers.

UsaProxy logs this information in a log file. This file consists of one entry per event that happens. During our tests, the size of this log file grew with approximately 30 kB per minute.

There were, however, a few drawbacks that we had to attend, to make the system useful for our research on children's search behaviour. First of all, UsaProxy could initially only measure events with an accuracy of a second. We have improved the accuracy of the system to one millisecond.

The second drawback was that UsaProxy is unable to record key presses on the delete and backspace buttons when using Internet Explorer. The user could delete letters from his query without UsaProxy noticing it. The number of words and the average length of words that we extract from the raw data is therefore an estimate. Firefox does detect these key presses. However, since we cannot be sure which browser is used when conducting experiments, we have chosen to disregard these key presses.

The third drawback was that the raw data the system delivers, was hardly usable for analysis. We needed to derive data from this that is more usable.

To solve this problem, we have written an application that takes a directory of log files and extracts useful information out of them. We have chosen to, at first, extract the following information from the log files. The level mentioned between parentheses is the level from section 4.2.2 that the variable belongs in.

1. Speed of typing (level 1)
2. Number of clicks per session (level 1)
3. Number of words per query (level 2)
4. Average length of words in the query (level 2)
5. Session length (level 2)

This set of variables is easily extractable from the log files and serves as a good starting point to assess the functionality of our application in this preliminary stage of our research.

We have defined a session to take place between consecutive queries. When a user returns to the start page to start a new search query, a new session starts. The information shown above is gathered and grouped per session.

Our application has been written in Java and can be run on any computer that has a Java Virtual Machine installed. The researcher can select a directory that contains log files which have been produced by UsaProxy. He can also enter what the start page is, so that the tool can identify where a new session starts.

The information that the application extracts from the log file is then stored in a comma separated values file. This file format is easily readable by Excel or other analysis applications.

In our research, we need to make sure that the information UsaProxy logs, is only generated by children using the websites, not by adults. This is where UsaProxy's flexibility proves its usefulness. We can install it anywhere, also on a computer that we know is only used by children.

The combination of an adapted version of UsaProxy and our own application, that extracts useful information out of UsaProxy's log files, is well suited for our research on children's search behaviour. We think that this combination may also be useful for researching search behaviour of adults, but finding evidence for this is not the aim of our research.

8. CONCLUSIONS

In this paper, we presented the development of an application that can monitor children's search behaviour on a large scale. Instead of examining problems children experience during information-seeking through high-quality research, this application gives us the opportunity to examine children's information-seeking problems through high-quantity research. The application can provide information about children's search behaviour on a much larger scale, that can be of high value for the research on developing search interfaces and search technologies that support children in effective and enjoyable information-seeking.

9. ACKNOWLEDGMENTS

This research is funded by the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement no. 231507, called 'PuppyIR'.

The research is also part of the research program called 'The digital youth library' that is funded by The Netherlands Public Library Association (VOB).

10. REFERENCES

[1] AlphaOmega Software, Keyboard and mouse recorder, http://alphaomega.software.free.fr/keyboardandmouserecorder/Keyboard%20And%20Mouse%20Recorder.html, retrieved 16 June 2009.

[2] Arroyo, E., Selker, T., and Wei, W. 2006. Usability tool for analysis of web designs using mouse tracks, In Proceedings of the CHI conference on Human Factors in Computing Systems (Montreal, Quebec, Canada, April 22-27, 2006).CHI'06. ACM Press, New York, NY 484 – 489.

[3] Atterer, R., Wnuk M., and Schmidt A. 2006. Knowing the user's every move – user activity tracking for website usability evaluation and implicit interaction, In Proceedings of the 15th international conference on World Wide Web (Edinburgh, Scotland, May 23-26). WWW'06. ACM Press, New York, NY, 203 – 212.

[4] Bar-Ilan, J. and Belous, Y. 2007. Children as Architects of Web Directories: An Exploratory Study. Journal of the American Society for Information Science and Technology, 58 (6), 895-907.

[5] Bilal, D. 2000. Children's use of the Yahooligans! web search engine: I. Cognitive, physical and affective behaviours on fact-based search tasks. Journal of the American Society for Information Science, 51 (7), 646 – 665.

[6] Bilal, D. and Kirby, J. 2002. Differences and similarities in information seeking: children and adults as Web users. Information processing and management, 38 (5), 649-670.

[7] Borgman, C.L., Hirsh, S.G., Walter, V.A. and Gallagher, A.L. 1995. Children's Searching behavior on browsing and keyword online catalogs: The Science Library Catalog Project. Journal of the American Society for information Science, 46 (9), 663-684.

[8] Broadbent, R.E., Saunders, G.S., and Ekstrom, J.J. 2006. An infrastructure for the evaluation and comparison of information retrieval systems, In Proceedings of the 7th conference on Information technology education (Minneapolis, Minnesota, USA, October 19-21, 2006). SIGITE'06. ACM Press, New York, NY, 123 – 127.

[9] Bruckman, A., Bandlow A., and Forte, A. 2008. HCI for Kids. In Handbook of Human-Computer Interaction, J. Jacko and A. Sears, Ed. Lawrence Erlbaum Associates, 793-809.

[10] Hirsh, S.G. 1999. Children's Relevance Criteria and information seeking on electronic resources. Journal of the American Society for Information Science, 50 (14), 1265-1283.

[11] Hong, J.I., Heer, J., Waterson, S., and Landay, J.A. 2001. WebQuilt: a proxy-based approach to remote web usability testing. ACM Transactions on Information Systems, 19 (3), 263 – 285.

[12] Inkpen, K. M. 2001. Drag-and-Drop versus Point-and-Click Mouse Interaction Styles for Children. ACM Transactions on Computer-Human Interaction, 8 (1), 1-33.

[13] M.G. van Kalsbeek and J.J. de Wit, Automatic reformulation of children's search queries, *Unpublished paper written for the M.Sc. course on Information Retrieval*, University of Twente, 2007.

[14] Muresan, G. and Bai, B. 2007. Exploring Interactive Information Retrieval: An Integrated Approach to Interface Design and Interaction Analysis. In Proceedings of the 8th International Conference on Computer-Assisted Information Retrieval (Pittsburgh, USA, May 30 – June 1, 2007). RIAO'07. ACM Press, New York, NY.

[15] Mueller, F. and Lockerd, A. 2001. Cheese: tracking mouse movement activity on websites, a tool for user modelling. In Proceedings of the the CHI conference on human factors in computing systems (Seattle, Washington, USA, March 31 – April 5, 2001). CHI'01. ACM Press, New York, NY, 279 – 280.

[16] Nicholas, D., Huntington, P., Jamali, H.R. and Watkinson, A. 2006. The information seeking behaviour

of the users of digital scholarly journals, Information Processing and Management: an International Journal, 42, 1345 – 1365.

[17] Nielsen, J. and S. Gilutz 2002. Usability of Websites for Children: 70 design guidelines based on usability studies with kids. J. Nielsen, Nielsen Norman Group.

[18] Piaget, J. 1970. Science of Education and the Psychology of the Child. New York: Orion Press.

[19] Read, J.C. and MacFarlane, S. 2006. Using the fun toolkit and other survey methods to gather opinions in child computer. In Proceedings of the 2006 conference on Interaction design and children (Tampere, Finland, June 7-9, 2006). IDC'06. 81 – 88.

[20] Rowlands, I. 2008. Information behaviour of the researcher of the future. British Library.

[21] Schacter, J., Chung, G.K.W.K., and Dorr, A. 1998. Children's internet searching on complex problems: performance and process analyses, Journal of the American Society for Information Science, 49, 840 – 849.

[22] Leiva Torres, L.A. and Hernando, R.V. 2007. (smt) Real time mouse tracking registration and visualization tool for usability evaluation on websites. In Proceedings of the IADIS international conference WWW/Internet (Vila Real, Portugal, October 5-8, 2007).

How Different are Wikipedia and Web Link Structure?

Jaap Kamps[1,2] Marijn Koolen[1]

[1] Archives and Information Studies, University of Amsterdam
[2] ISLA, Informatics Institute, University of Amsterdam
{kamps,m.h.a.koolen}@uva.nl

ABSTRACT

In this paper, we investigate the difference between Wikipedia and Web link structure with respect to their value as indicators of the relevance of a page for a given topic of request. Our main findings are: First, Wikipedia link structure is similar to the Web, but more densely linked. Second, Wikipedia's outlinks behave similar to inlinks and both are good indicators of relevance, whereas on the Web the inlinks are more important. Third, when incorporating link evidence in the retrieval model, for Wikipedia the global link evidence fails and we have to take the local context into account.

1. INTRODUCTION

The principal difference between Web retrieval and general information retrieval, is the abundant link structure of the Web which can been exploited to improve information retrieval in algorithms [4, 7]. Similar to the earlier use of citations in bibliometrics, a link can be considered as a "vote" for a page being authoritative. Wikipedia's links are a special case of the general hyperlinks that connect the World Wide Web. Internal links in Wikipedia are typically based on words naturally occurring in a page and link to another "relevant" Wikipedia page. Our conjecture is that the links in Wikipedia are different from links between arbitrary Web documents.

Our main research question is to find out if, and how, the link structure of Wikipedia differs from the Web at large with respect to its value for promoting retrieval effectiveness. To investigate this, we use two IR test collections consisting of documents plus search requests and associated relevance judgments. For Wikipedia, we use the INEX 2006 and 2007 Ad hoc collections, together consisting of 217 ad hoc topics and an XML version of Wikipedia containing over 650,000 articles [1]. and for the Web we use the TREC 2004 Web Track collection, consisting of 225 topics and the 1.2 million documents .GOV collection. We make no particular claims on the representativeness of this data set for the current Web, which is infinitely large and highly heterogeneous, but expect it to be a close enough approximation for our purposes [8].

Our main research question breaks down in two parts. We start by investigating the Wikipedia link structure with a comparative analysis of the two IR test collections, Wikipedia and .GOV. The second part of our main research question is about the effectiveness of link-based evidence. At TREC, we have seen that link degree is not effective for general ad hoc retrieval [2]. However, for web-centric retrieval tasks like entry page finding, link indegree proved highly beneficial [5]. What is the impact of link evidence on Web-centric retrieval on .GOV and ad hoc retrieval on Wikipedia?

Table 1: Statistics of the .GOV and Wikipedia collections

		min	max	mean	median	stdev
GOV	Indegree	0	44,228	8.90	1	126.00
	Outdegree	0	653	8.90	4	16.61
Wiki	Indegree	0	74,937	20.63	4	282.94
	Outdegree	0	5,098	20.63	12	36.70

2. COMPARATIVE ANALYSIS

In this section, we look in detail at the link structures of the Wikipedia and Web collections. The .GOV collection contains 1,247,753 documents and 11,110,989 unique links between these pages (we ignore links which point to, or from, pages outside the collection). The Wikipedia collection contains 659,304 documents and a total of 13,602,613 unique links between these pages. We have also looked at how many of these links are reciprocal: there are 1,269,988 (11.4%) reciprocal links in the .GOV collection, and 1,182,558 (8,7%) reciprocal links in the Wikipedia collection. The higher fraction of reciprocal links in the .GOV collection is likely due to the presesence of nagivational links within web-sites. Statistics of the degree distributions is given in Table 1. The Wikipedia collection has fewer documents and a larger number of links and is thus more densely linked. This is surprising in the sense that the .GOV domain is much older, and link density tends to increase over time [6]. There are two effects which help explain why the Wikipedia link graph is more "complete" than the .GOV link graph. First, due to the structured nature of Wikipedia, it is much clearer for Wikipedia authors where to link to. Second, due to peer editing and automatic link detection, "missed" links will be added over time.

We analyse the prior probability of relevance (PoR) of a page with a particular link degree. We use IR test-collections with search topics and associated relevance judgments. For the 225 topics of the .GOV collection we have 1,763 relevant documents, for the 217 topics of the Wikipedia collection we have 11,896 relevant documents. If the degrees of relevant documents deviate from the degrees of non-relevant documents, they may possibly be used as indicators of relevance. We calculate the PoR as follows. We sort all documents on ascending degree into bins of 10,000 documents. The PoR for the documents in each bin is the ratio of relevant documents in that bin. If link degree is related to relevance, we expect the PoR to go up with increasing degree. Figure 1 shows the results. In the .GOV collection, the probability of a document being relevant increases with indegree. For outdegree, the probability of relevance initially rises but then drops as the outdegree further in-

*This is an extended abstract of: J. Kamps and M. Koolen. Is Wikipedia link structure different? In *Proceedings WSDM 2009*, pages 232–241. ACM, 2009.

DIR-2010 January 25, 2010, Nijmegen, the Netherlands.

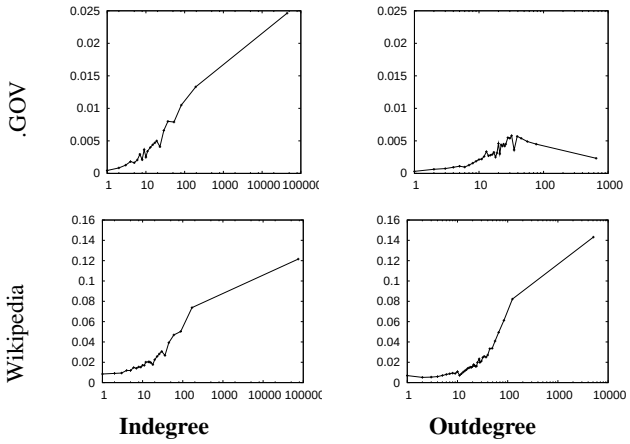


Figure 1: Prior probability of relevance of indegree (left) and outdegree (right) for .GOV (top) and Wikipedia (bottom)

creases. In the Wikipedia collection both in- and outdegree seem to be good indicators of relevance: a higher degree corresponds to a higher probability of relevance. This is not a result of pages linking back-and-forth, the fraction of reciprocal links in Wikipedia is actually lower than in .GOV. This suggests that outlinks in Wikipedia behave very much like inlinks. This is consistent with a semantic nature of links in Wikipedia: if a link from A to B means that B is relevant (in some sense) to A, then it is also likely A is relevant (in some sense) to B. This signals differences in the link structure of Wikipedia and the Web at large. For the semantic links of Wikipedia, the difference between incoming and outgoing links seems to disappear and both can be used as indicators of relevance.

3. EFFECTIVENESS OF LINK EVIDENCE

We work in the language modelling framework, incorporating link evidence into the retrieval model similar to Kraaij et al. [5]. We multiply the content-based retrieval score with the link degree and conduct experiments with them on the TREC 2004 Web track topics and on the combined INEX 2006 and 2007 Ad Hoc track topics. Link indegree can be considered on a global level, i.e. indegree over the whole collection (similar to PageRank), or on a local level, i.e. indegree within the subset of articles retrieved as results for a given topic (similar to HITS). For the local link degrees we use only the links between the top 100 ranked results.

Results for the .GOV collection are shown in Table 2. As we expected from the PoR plots, the indegrees are much more effective than the outdegrees, although the outdegrees are still effective. The local degrees are more effective for Mean Average Precision (MAP), but the global outdegrees are the most effective for Mean Reciprocal Rank (MRR). Taking the log of the priors to tone down their impact is less effective.

Results for the Wikipedia collection are shown in Table 3. Here, both the global in- and outdegrees improve MRR but hurt MAP, even when logged. For ad hoc retrieval, with many relevant documents, global link evidence leads to infiltration of important but off-topic pages that are ranked low on content score. Local link degrees lead to significant improvements, with little difference between the impact of in- and outdegrees.

4. DISCUSSION AND CONCLUSIONS

We investigated the difference between Wikipedia and Web link structure, based on evidence from two IR test-collections. Wikipedia is more densely linked than .GOV. We observe that Wikipedia

Table 2: Results of the different link priors over in- and outdegree on the 225 topics of the Web track collection

	MAP		MRR	
Run id	**Glob**	**Loc**	**Glob**	**Loc**
baseline	0.3970		0.4662	
in	0.4738$^{\bullet}$	**0.4799**$^{\bullet}$	**0.5885**$^{\bullet}$	0.5655$^{\bullet}$
out	0.4299$^{\circledcirc}$	0.4497$^{\bullet}$	0.5046$^{\bullet}$	0.5199$^{\bullet}$
log.in	0.4449$^{\bullet}$	0.4410$^{\bullet}$	0.5209$^{\bullet}$	0.5148$^{\bullet}$
log.out	0.4082$^{\circledcirc}$	0.4181$^{\bullet}$	0.4789$^{\bullet}$	0.4879$^{\bullet}$

Table 3: Results of the different link priors over in- and outdegree on the 217 topics of the Wikipedia collection

	MAP		MRR	
Run id	**Glob**	**Loc**	**Glob**	**Loc**
baseline	0.3090		0.8121	
in	0.3018^{-}	0.3190$^{\circledcirc}$	0.8139^{-}	0.8236^{-}
out	0.3016^{-}	**0.3199**$^{\bullet}$	0.8262^{-}	0.8266^{-}
log.in	0.2865^{-}	0.3176$^{\bullet}$	**0.8322**$^{\circ}$	0.8289°
log.out	0.2890^{-}	0.3156$^{\bullet}$	0.8291$^{\circledcirc}$	0.8225°

inlinks and outlinks are similar in character, leading to the conflation of the notions of authority and hub [4].

In our retrieval experiments, we wanted to know what the impact is of link evidence on retrieval. For the Web track collection, all global and local outdegree priors are less effective than the corresponding indegree priors, supporting the claim that document importance is a major aspect in Web retrieval. Global indegree is more effective for early precision, which is important for Web search.

For the Wikipedia collection, the outdegree priors behave very similar to the indegree priors. The brute force of the global degree priors is too much for the task of ad hoc retrieval. Even the more subtle log degree prior is not effective for MAP. The local degrees stay more on topic and can improve early and later precision, showing that link evidence has to be carefully weighted and made sensitive to the local context.

Acknowledgments This research was supported by the Netherlands Organization for Scientific Research (NWO), CATCH programme, under project number 640.001.501.

REFERENCES

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

[2] D. Hawking. Overview of the TREC-9 web track. In *TREC*, 2000.

[3] J. Kamps and M. Koolen. Is Wikipedia link structure different? In *Proceedings WSDM 2009*, pages 232–241. ACM, 2009.

[4] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[5] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings SIGIR 2002*, pages 27–34. ACM, 2002.

[6] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings KDD '05*, pages 177–187. ACM, 2005.

[7] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[8] I. Soboroff. Do trec web collections look like the web? *SIGIR Forum*, 36:23–31, 2002.

Dutch Parliamentary Debates on Video

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

ABSTRACT

We created an archive of video footage of the meetings of the Dutch parliament. All video was aligned with the official transcripts (a.k.a. Hansards or proceedings) of these meetings. A prototype search interface was built. The paper describes the data and the main technical aspects of the project.

Categories and Subject Descriptors

H.4.m [**Information Systems**]: Miscellaneous

Keywords

Dutch, Video, Politics

1. OBJECTIVE

Starting with the Scottish Parliament [5], several legislative bodies provide information systems giving access to video footage of their plenary meetings. The most advanced systems have segmented the video into natural units. Typically these are the topics discussed at the meeting, or, more fine-grained, the speeches made by members of the parliament or council. The same segmentation is made in the verbatim proceedings of the meeting and the two media are aligned.

This setup yields a richly annotated video data set for which entry point retrieval can be implemented using rather standard IR systems. Techniques from XML retrieval systems [3, 6] apply well for this use case. A well-designed example is the site `http://theyworkforyou.com` created by MySociety.

The objective of our work was to create a similar system for the Dutch Parliament. The availability of Dutch parliamentary proceedings is rather complicated:

- data from before 1995 is available from the Royal Dutch library at `http://statengeneraaldigitaal.nl`;
- data from 1995 to the present is available from the SDU, the former state printer, through the Parlando website;
- recent data (less than three weeks old) is available in HTML format from the website of the Dutch Parliament `http://www.tweedekamer.nl`;
- video data is not archived and was at the time we started this research not systematically stored. Data is streamed live on the web at `http://www.tweedekamer.nl`.

The PoliDocs parliamentary information system [1] provides a single access point to all parliamentary data in one uniform XML format. Proceedings are segmented into topics and blocks of speeches with interruptions. These blocks are divided into speeches by speakers, which are again divided into paragraphs. For every word being said in parliament, it is thus explicitly coded in the XMLmarkup who said it, in what context, and when. An entry point retrieval system which returns speeches is available at `http://polidocs.nl`.

Our aim was to extend each search result in that system with a link to the exact point in the video of the meeting in which the found speech was made. A proof of concept was built and can be seen at `http://openkamer.unwind.nl`. Based on that the Dutch ministry of internal affairs through the Digital Pioneers foundation awarded a subsidy to built a prototype which is available at `http://openkamer.tv`.

We briefly describe some of the technical aspects of this system. Figure 1 contains a screen shot.

Data format.

The video data is stored in two formats: Windows Media Video (source format) and Adobe Flash Video. The Flash Video format is obtained from the Windows Media Video using FFmpeg. The videos come with the following metadata: ID, date , time and duration. Through the ID each video is linked to the proceedings in XML format. Each speech in the XML file contains an attribute with a timecode.

Data collection.

We collected video data since 1 March 2009. At the time of writing (December 2009), the video corpus comprises 4190 hours, taken from 401 meetings (both plenary and committee) held at 117 days. The corpus occupies 838 GB of disc space.

Downloading video material from a webstream inevitably leads to some data loss. We use a script which quickly recuperates after a break in the transmission. On average the script picks up the stream within less than a minute. We estimate that less than 4 % of the transmitted video is lost. For a daily overview see `http://www.openkamer.tv/status.php`.

DIR2010 Nijmegen

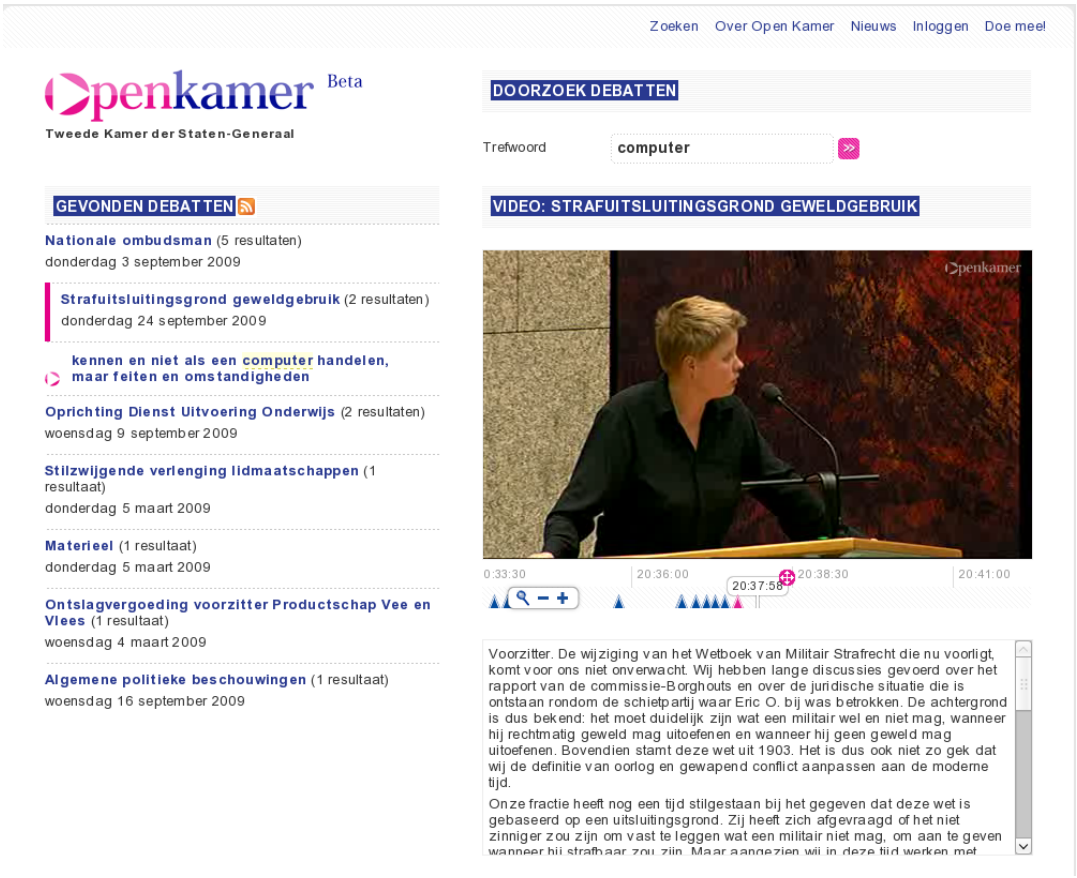


Figure 1: Screen shot of the system at `http://openkamer.tv`

Aligning text and video.

The main problem in the project was the alignment of video and text. As we had segmented the text already at the level of speakers all we needed was to segment the video into speakers as well. We considered the following four techniques for this:

1. Obtain explicit timecodes from the stenographical section ofhe Dutch Parliament. This turned out to be impossible.

2. Every speaker change in the Dutch parliament involves a change in the used microphone (chairmain, central lectern, and interruption microphone). Speaker segmentation would thus have been easy if we had access to the multi-track audio tape. This turned out to be impossible.

3. Speaker segmentation (also called diarization) using the audio and text files. We tried two systems [4, 2] but the results were not good enough for a live system.

4. Manual segmentation. We created a web based alignment tool with which the speaker segmentation and alignment of proceedings and video can be performed. The tool works smoothly. After a short training our coders could align 200 hours of video in 120 hours.

Acknowledgements

Maarten Marx acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599.

De ontwikkeling van Openkamer.tv is mogelijk gemaakt door een bijdrage van Kennisland in de regeling Digitale Pioniers ronde eParticipatie (die door het ministerie van Binnenlandse Zaken en Koninkrijksrelaties genitieerd is). Thanks are due to Steven Grijzenhout and Ernst van Rheenen for providing scraper scripts.

2. REFERENCES

[1] T. Gielissen and M. Marx. Exemelification of parliamentary debates. In *Proceedings DIR 2009*, pages 19–25, 2009.

[2] M. Huijbregts. *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled.* Phd thesis, Univ. of Twente, 2008.

[3] C. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[4] A. Noulas and B. Krose. On-line multi-modal speaker diarization. In *Proceedings ICMI '07*, pages 350–357, 2007.

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

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

DutchParl
A Corpus of Parliamentary Documents in Dutch

Maarten Marx and Anne Schuth
ISLA, University of Amsterdam
Kruislaan 403 1098 SJ Amsterdam, The Netherlands
maartenmarx@uva.nl aschuth@science.uva.nl

ABSTRACT

A corpus called DutchParl is created which aims to contain all digitally available parliamentary documents written in the Dutch language. The first version of DutchParl contains documents from the parliaments of The Netherlands, Flanders and Belgium. The corpus is divided along three dimensions: per parliament, scanned or digital documents, written recordings of spoken text and others. The digital collection contains more than 800 million tokens, the scanned collection more than 1 billion.

All documents are available as UTF-8 encoded XML files with extensive metadata in Dublin Core standard. The text itself is divided into pages which are divided into paragraphs. Every document, page and paragraph has a unique URN which resolves to a web page. Every page element in the XML files is connected to a facsimile image of that page in PDF or JPEG format. We created a viewer in which both versions can be inspected simultaneously. A search-engine for the complete collection is available online.

The corpus is available for download in several formats. The corpus can be used for corpus-linguistic and political science research, and is suitable for performing scalability tests for XML information systems.

Keywords

Dutch, Text corpus, Politics, XML

1. INTRODUCTION

The main reason to create the corpus is to provide one portal from which these documents are accessible both in their original official version (in PDF format), and in a uniform XML format with extensive metadata [2]. The corpus was designed to be useful as a data set in all possible scientific disciplines. E.g., it can be used for (comparative) corpus-linguistic and political science research and as a test-set for information-theoretic experiments. This distinguishes DutchParl from EuroParl [1] which is developed for research in Statistical Machine Translation. The corpus was developed following the guidelines set out in [2].

How to get the corpus?

The corpus is available for download at

`http://politicalmashup.nl/DutchParl`.

A full version of this paper is also available there. We are not aware of copyright restrictions on the material. If you use the corpus, please sent an email to `maartenmarx@uva.nl`.

2. COVERAGE AND SIZE OF DUTCHPARL

Spatial and temporal coverage.

Parliamentary documents in the Dutch language are produced in the following locations: Belgium (Flemish parliament, and the Belgian federal parliament), European Union, Suriname and The Netherlands. The present version of DutchParl does not yet contain data from the EU nor from Suriname.

The periods for which data is available differ per source. Table 1 lists the periods for which digital and scanned data is available on the web for each source (measured in September 2009). This is exactly the data available in DutchParl.

Subcorpora.

The corpus can be divided into many subcorpora. This is facilitated by the uniform metadata using a controlled vocabulary. In the description below we partition the data along three dimensions. First by source: Belgium, Flanders and The Netherlands. Secondly, digitally produced documents are separated from scanned and OCR-ed documents. The latter contain noise in the form of wrongly recognized characters, mistakes in paragraph splitting, non UTF-8 characters, or simply no extractable text.

A special subset of the parliamentary documents are the verbatim notes of sessions of parliament. Even though the texts are edited and transcribed to be read, they are accounts of spoken language. For this reason, we present details both for the complete collections and for the verbatim notes separately.

Size of DutchParl.

Table 2 displays information about the size of the subcorpora. We note that the documents from the Belgian parliament are bilingual, with text in Dutch and French interspersed in many different ways.

Source	Digital	OCR-ed	Planned
Belgium	From 1999-07-01	-	1844–1999 is scanned
Flanders	From 1995-10-17	1971-12-07 to 1995-10-17	-
The Netherlands	From 1995-01-01	1917-01-01 to 1995-01-01	1814–1917 available in 2010

Table 1: Availability of parliamentary data in the Dutch language.

Subcorpus	Mbyte text	# Documents	# Pages	# Tokens
Belgian Federal	800	3.901	216.522	129.085.483
Flanders	454	5.470	161.881	72.958.408
Netherlands	4.331	198.433	1.594.845	684.932.669
Flanders OCR	146	1.018	34.867	23.924.567
Netherlands OCR	7.043	328.722	1.701.130	1.003.555.596

Subcorpus	Mbyte text	# Documents	# Pages	# Tokens
Belgian	502	3.462	137.366	81.086.575
Flanders	311	3.799	93.591	50.715.218
Netherlands	781	21.604	137.610	131.681.453
Flanders OCR	142	932	33.147	23.378.215
Netherlands OCR	2.644	12.796	383.863	402.657.396

Table 2: Number of documents, pages and tokens for the complete corpus (top) and only for verbatim notes of parliamentary and committee sessions (bottom).

	NL-DIGITAL	NL-SCAN	Flanders DIGITAL	Flanders SCAN	BE-federal
Total number of words	102870201	329540359	38629223	17120704	41152224
Unique words	353677	1963712	258304	184945	245447
Words occurring just once	149719	1311243	118992	91889	102093
Words occurring more than once	203958	652469	139312	93056	143354
Words occurring at least 4 times	130008	370932	88518	57277	90911
Words occurring at least 20 times	55054	134735	36413	22945	37250

Table 3: Token counts; all data (top) and verbatim notes of parliamentary sessions (bottom).

Number of tokens.

Table 3 presents figures on the number of tokens occurring in the different subcorpora. Again we make a distinction between digital and scanned documents and present the numbers for the spoken texts separately.

3. CONCLUSIONS AND FUTURE WORK

This work started out as a challenging data integration project: Can we collect and bring together under one uniform schema parliamentary data from different countries, produced in different periods of time, and available in different formats? DutchParl showed that we partly succeeded. We created a rich metadata schema based on Dublin Core standards. However, it is not always easy or possible to collect meaningful data for all fields (we did not manage for Belgium Federal). Also, even after many tries and promises, we did not receive the data from Suriname. A hard problem is checking completeness. Even if we are confident that we downloaded all material available on the web, we cannot be sure that we have all material. It is difficult to find reliable independent listings of material.

We paid extra care to providing provenance information. Because we assigned corpus unique ID's to every paragraph, page and document, specific referencing of material (common in the social sciences) is possible using hyperlinks. The connection of the data in XML with the original official publications is quite specific and convenient because we provide a facsimile image of every page.

Future challenges include 1) keeping the corpus daily up to date, 2) managing the data in an XML database managment system, 3) scaling to other countries, 4) linking the data with other datasets, e.g. bibliographies of MP's, 5) performing text analytics on noisy data, 6) machine translation, 7) create a search system.

Acknowledgements

Maarten Marx acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599.

4. REFERENCES

[1] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proc. MT summit*, volume 5, 2005.

[2] M. Wynne. Archiving, distribution and preservation. In M. Wynne, editor, *Developing Linguistic Corpora: a Guide to Good Practice*, pages 71–78. Oxford: Oxbow Books, 2005. Available online from `http://ahds.ac.uk/linguistic-corpora` [Accessed 2009-07-01].

Learning Semantic Query Suggestions*

Edgar Meij
University of Amsterdam
Science Park 107
Amsterdam, The Netherlands
edgar.meij@uva.nl

Marc Bron
University of Amsterdam
Science Park 107
Amsterdam, The Netherlands
m.m.bron@uva.nl

Laura Hollink
VU University Amsterdam
de Boelelaan 1081a
Amsterdam, The Netherlands
hollink@cs.vu.nl

Bouke Huurnink
University of Amsterdam
Science Park 107
Amsterdam, The Netherlands
bhuurnink@uva.nl

Maarten de Rijke
University of Amsterdam
Science Park 107
Amsterdam, The Netherlands
derijke@uva.nl

* Human-defined concepts are fundamental building blocks of the semantic web. When used as annotations for documents or text fragments they can provide explicit anchoring in background knowledge and enable intelligent search and browsing facilities. As such, an important application of ontological knowledge is augmenting unstructured text with links to relevant, human-defined concepts. For the author or reader of the text, this augmentation may supply useful pointers, for example to the concepts themselves or to other concepts related to the ones found. For ontology learning applications, such links may be used to learn new concepts or relations between them [11]. Recently, data-driven methods have been proposed to generate links between phrases appearing in full-text documents and a set of ontological concepts known a priori. Mihalcea and Csomai [8] propose the use of several linguistic features in a machine learning framework to link phrases in full-text documents to Wikipedia articles and this approach is further improved upon by Milne and Witten [9]. Because of the connection between Wikipedia and DBpedia, such data-driven linking methods help us establish links between textual documents and Linked Open Data [1, 2, 4, 10].

Another, more challenging instantiation of linking text to human-defined concepts in a knowledge source is *semantic query suggestion*. Query suggestion is a strategy to derive terms that are able to return more relevant results than the initial query. Commonly used approaches to query suggestion (sometimes referred to as a form of query expansion) are highly data-driven and based mostly on term frequencies [5, Chapter 9]. *Semantic* query suggestion, in contrast, tries to understand (or learn) which concepts the user used in her query or, phrased alternatively, the concepts she is interested in and wants to find.[1] Moreover, the properties of each concept, and any other resources associated with it, could serve as additional, useful information for the user. In our current work, we use DBpedia as our target ontology. As an example of our task, consider the query "obama white house". A semantic query suggestion algorithm should return suggestions in the form of the (DBpedia) instances labeled "Barack Obama" and "White House". Identifying such semantic suggestions serves multiple purposes: it can (i) help the user acquire contextual information, (ii) suggest related concepts or associated terms that may be used for search, and (iii) provide valuable navigational suggestions.

In this paper we address the semantic query suggestion task and automatically link queries to DBpedia concepts. The specific task we address in this paper is the following. Given a query that is submitted to a search engine, identify the relevant concepts that the user entered in her query where the concepts are taken from an existing knowledge base or ontology. We address our task in the setting of a digital archive, specifically of the Netherlands Institute for Sound and Vision ("Sound and Vision"). Sound and Vision maintains a large digital audiovisual collection, currently containing over a million objects and daily updated with new broadcasts.

Our approach to suggesting DBpedia concepts for user queries consists of two stages. In the first stage, a ranked list of possible concepts for the query is generated using a language modeling framework for each full query and for each n-gram (i.e., contiguous sequence of n words) in the query. We use various textual representations of each DBpedia concept, including the Wikipedia article text, its label, and the text used in the hyperlinks pointing to it.

Once we have obtained a ranked list of possible concepts for each n-gram in the query, we turn to concept selection. In this stage we need to decide which of the candidate concepts are most viable. We use a supervised machine learning approach, which takes as input a set of labeled examples (query to concept mappings) and several features of these examples. We choose to compare a Naive Bayes (NB) classifier, with a Support Vector Machine (SVM) classifier and a decision tree classifier (J48)—a set representative of the state-of-the-art in classification. We experiment with multiple classifiers in order to confirm that our results are generally valid, i.e., not dependent on any machine learning algorithm. In order to

*This an extended abstract of Meij et al. [7].

[1]We use "ontology" to refer to the full spectrum of conceptualizations, ranging from glossaries to formal ontologies [6]. We refer to an instance in DBpedia as "concept" [10].

train the machine learning algorithms, we examined close to 1000 queries from a search engine for a digital multimedia archive and manually linked over 600 of these to relevant concepts in DBpedia. We employ several types of features, each associated with either the current query n-gram, the current concept, their combination, or the session history.
We define semantic query suggestion as a ranking problem; the system has to return five concepts for a given input query and the assessments described above are used to determine the relevance status of these concepts. We employ several measures which are well-known in the field of information retrieval [3].
Using Support Vector Machines and features extracted from the full input queries yields optimal results. The best performing run is able to locate almost 90% of the relevant concepts on average. Moreover, this particular run achieves a precision@1 of 89% which means that for this percentage of queries a relevant concept is returned as the first suggestion.
In sum, we have shown that the semantic query suggestion problem can be successfully cast as a ranking problem. The best way of handling query terms is not as separate n-grams, but as a single unit—a finding also interesting from an efficiency viewpoint, since the number of n-grams is quadratic with respect to the length of the query. All types of feature were found to be helpful and, besides document and term features, we found that concept features were also important in achieving our best performance.

Acknowledgments

This research was carried out in the context of the Virtual Laboratory for e-Science project and supported by the DuOMAn project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project number STE-09-12 and the DAESO project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (http://www.stevin-tst.org) under project number STE-05-2 and the Netherlands Organisation for Scientific Research (NWO) under project numbers 017.001.190, 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802.

References

[1] S. Auer and J. Lehmann. What have Innsbruck and Leipzig in common? Extracting semantics from wiki content. In *The Semantic Web: Research and Applications*, 2007.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. *ISWC '07*, 2007.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[4] C. Bizer, R. Cyganiak, S. Auer, and G. Kobilarov. DBpedia–querying Wikipedia like a database. In *WWW '07*, 2007.

[5] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[6] D. L. Mcguinness. Ontologies come of age. In D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003.

[7] E. J. Meij, M. Bron, B. Huurnink, L. Hollink, and M. de Rijke. Learning semantic query suggestions. In *8th International Semantic Web Conference (ISWC 2009)*, 2009.

[8] R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM '07*, 2007.

[9] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM '08*, 2008.

[10] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW '07*, 2007.

[11] W. R. van Hage, M. de Rijke, and M. Marx. Information retrieval support for ontology construction and use. In *ISWC '04*, 2004.

More! A Social Discovery Tool for Researchers

Gonzalo Parra
Katholieke Universiteit Leuven
Celestijnenlaan 200 A
B-3001, Heverlee, Belgium
+3216327660
gonzalo.parra@cs.kuleuven.be

Sten Govaerts
Katholieke Universiteit Leuven
Celestijnenlaan 200 A
B-3001, Heverlee, Belgium.
+3216327552
sten.govaerts@cs.kuleuven.be

Erik Duval
Katholieke Universiteit Leuven
Celestijnenlaan 200 A
B-3001, Heverlee, Belgium
+3216327066
erik.duval@cs.kuleuven.be

EXTENDED ABSTRACT

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering
D.2.2 [**Design Tools and Techniques**]: User interfaces

General Terms

Design, Experimentation, Human Factors.

Keywords

social discovery tools, science 2.0, human computer interaction, user interfaces, mobile devices, ubiquitous computing

1. INTRODUCTION

"Science 2.0" is the result of "Web 2.0" tools and trends influencing the research area [1]. The effects are visible on how researchers experiment, get feedback on their work, and interact with their community [2][3]. We focus on the scenario where a researcher is attending a conference presentation and is interested in the topic and the speaker: it comes as a natural action to try to find more information about them. Currently this can be done in an ad hoc way, where the researcher either talks to the speaker or uses a search engine to find information about the speaker, his home page, blog, publications list, etc. A big drawback is that this process is far from instantaneous: the attendee may search during the conference session, or write down that he wants to follow-up later on. Oftentimes, this follow-up doesn't happen. And even if it does, it may no longer be as useful as the attendee may no longer be at the conference and it may be difficult to find the relevant information to begin with.

Thus, we identified a need to easily find speaker information during the presentation, and to subscribe to feeds that keep the attendee informed about ongoing work from the speaker. We have addressed this need through a mobile application, called "More!".

The structure of this extended abstract is as follows: section 2 discusses the design and implementation of the tool proposed to solve the need mentioned above. Section 3 presents an evaluation of the usability and functionality of the tool. Finally, we include conclusions and opportunities for further work.

2. THE MORE! APPLICATION

We have developed a web application that groups relevant information of a speaker in a way that can be easily exposed and integrated in the normal workflow of the audience.
The application provides:

- mobility, so that it can be used at the conference;
- a unique fingerprint to identify the speaker, and
- relevant information about the speaker.

The mobile web application is called "More!" as it provides more information about the speaker. We were inspired by the Shazam application that provide similar functionality for music [4]. "More!" is a web application that is optimized for viewing on a mobile device, but can also be used from a regular computer with a web browser. Quick Response (QR) codes were selected to represent the speaker fingerprint [5]. The QR code encodes a URL that resolves to the speaker page on the "More!" web application.

Finally, we expose the following information for each speaker:

- speaker: full name, photo, electronic mail, affiliation and previous publications.
- current presentation: slides and paper.
- social tools: Twitter, SlideShare, blog, Delicious and Facebook.

In this way, the attendee can access some personal details about the speaker, as well as the paper and slides of the current presentation. Moreover, he can 'follow' the speaker on some of the more mainstream "Web 2.0" social tools, in order to stay informed of new work by the speaker.
The workflow of the application is as follows:

1. The speaker exposes a QR code, either on the first slide or on all the slides of his presentation.
2. Attendants capture and decode the QR code with their smart phones and are redirected to the "More!" application.
3. "More!" obtains the data and presents it on the client tool.

An initial paper mockup of the application was developed and evaluated with three potential final users (researchers). Regarding the contents of the application, only one user suggested to include more information about the speaker, in this case an extra social tool. On the usability side, all of the participants suggested to either include more features to the application or to expose the options in a different way. Based on this feedback, we developed a final design that did not include any extra features but we did re-arrange the presentation of the information. The final application could be seen in Figure 1.

3. EVALUATION

Two evaluations took place using the implemented version of the application. Different scenarios were used, to evaluate both usability and functionality of the application.

3.1 Initial Evaluation

The usability of the tool was evaluated by empirical and analytic methods [6][7][8]. This evaluation took place with 20 university students of Computer Science at the K.U.Leuven. The students were presented with a description of the application and a typical scenario where it could be applied. Smart phones were provided for the evaluation. It is important to mention that the group of students had already prior knowledge of social tools, but only basic or no experience with mobile devices.

Figure 1. Final design of the "More!" application

All participants agreed that the tool enabled them to obtain more information about the speaker. Regarding the presentation of the current work, 84% agreed that "More!" helps them to follow the presentation, and the rest weren't sure. About the simplicity of the application, the students agreed that "More!" is simple to use and they are fairly satisfied with the experience.

The students were requested to list the strengths and weaknesses of the application. The biggest strength was the use of the graphical code as the initial fingerprint and the possibility to explore the slides. The major weakness was the incompatibility of some social tools with the smart phone web browser.

The major problem encountered with this evaluation was related to the devices. These were not used regularly by the students and in some cases not properly configured (e.g. no mail account configured).

3.2 First evaluation in practice

The second evaluation of the tool focused on the functionality and took place in a real world situation. The 15 participants of a workshop at the Alpine Rendez-Vous 2009 [9] were introduced to the experiment and requested to use the tool. The use of the application was tracked and the participants were requested to fill out a questionnaire.

The preliminary results allow us to conclude that the tool was successfully accepted and used among the users. During the days the workshop took place there were 42 visits to the web application, where 19 where unique visitors. In total there were 97 page views with an average of 4 minutes per visit. While Mobile Safari was the most used mobile web browser with almost 53%, regular desktop clients like Firefox, Opera and the desktop version of Safari, were also used. This indicates that avoiding the dependence on a smart phone technology was a correct decision, in order to encourage more participants to use the application.

The participants clearly agreed that the application is simple and easy to use, with an average agreement of 4,75 over 5. The participants were a bit less satisfied with the design of the tool. Regarding the functionality, the participants agreed, with an average of 4,25 points, that the tool enabled them to obtain more information about the speaker.

4. CONCLUSIONS

There is a big interest in the research community for "Science 2.0" tools that can improve the way research is done.

From the Human Computer Interaction point of view, the users agreed that the tool is simple to use and are fairly satisfied with the components and design of the tool. Also in a real life setting, the users expressed that the tool helps them to know more about the speaker and easily follow his presentation. The web application approach of the tool proved to be useful as attendees used both smart phones browsers and regular web browsers.

5. FURTHER WORK

"More!" could improve the connections between researchers and there is plenty of opportunity for further improvement. Currently, there is some initial work regarding automatic extraction the information from the scientific papers in PDF. The reference-parsing package ParsCit [10] was used to obtain relations between authors, e-mails and affiliations. Then the Levenshtein text distance metric [11] was applied to match the extracted results. The main problems were related with duplicate authors, e-mail addresses structure, and notations of affiliations and mapping these back to the authors. The quality and scope of the data can be improved by using tools like DBLP[12] and Linked Data approaches [13], or by direct contacts with the publishers in order to increase the coverage of the database.
On the other hand, the evaluation participants expressed the need to manipulate the exposed the application's data. This modify/update feature should be considered.

In any case, applications like "More!" and the evolution to "Science2.0" represents a huge potential for improving the effectiveness and efficiency of our community.

6. REFERENCES

[1] Shneiderman, B. 2008., "Science 2.0". Science 319 (5868), 1349. DOI= 10.1126/science.1153539

[2] Waldrop, M. M. 2008. Science 2.0 - is open access science the future?. Scientific American. [Online]. Available: http://www.sciam.com/article.cfm?id=science-2-point-0

[3] Reinhardt, W., Ebner M., Beham G., and Costa, C. 2009. How people are using twitter during conferences. In Procs. of the 5th EduMedia conference, 145–156.

[4] Shazam web site: http://www.shazam.com/

[5] Q-R code web site: http://www.qrcode.com/

[6] Hartson, H.R., Andre, T.S., and Williges, R.C. 2003. Criteria for evaluating usability evaluation methods. Int'l Journal of Human-Computer Interaction 15(1), 145–181.

[7] Nielsen, J. 1993. Usability engineering. Academic Press, Boston.

[8] Rangel De Queiroz, J. E. and Sousa Ferreira, D. 2009. A Multidimensional Approach for the Evaluation of Mobile Application User Interfaces. In Procs. of the 13th Int'l Conf. on Human-Computer interaction, 242-251. DOI= http://dx.doi.org/10.1007/978-3-642-02574-7_27

[9] http://www.stellarnet.eu/programme/wp3/rendez-vous/

[10] Councill, I. G., Lee Giles, C. and Kan M. 2008. ParsCit: An open-source CRF reference string parsing package. In Procs. of Language Resources and Evaluation 08.

[11] Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8.

[12] http://dblp.uni-trier.de/

[13] Linked Data web site: http://linkeddata.org

Cross-Media Alignment of Names and Faces

[Extended Abstract] *

Phi The Pham
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
Heverlee B-3001, Belgium
PhiThe.Pham@cs.kuleuven.be

Marie-Francine Moens
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
Heverlee B-3001, Belgium
Sien.Moens@cs.kuleuven.be

Tinne Tuytelaars
Department of Electrical Engineering
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10
Heverlee B-3001, Belgium
Tinne.Tuytelaars@esat.kuleuven.be

ABSTRACT

In this paper we report on our experiments on aligning names and faces as found in images and captions of online news websites. Developing accurate technologies for linking names and faces is valuable when retrieving or mining information from multimedia collections. We perform exhaustive and systematic experiments exploiting the (a)symmetry between the visual and textual modalities. This leads to different schemes for assigning names to the faces, assigning faces to the names, and establishing name-face link pairs. On top of that, we investigate generic approaches to the use of textual and visual structural information to predict the presence of the corresponding entity in the other modality. The proposed methods are completely unsupervised and are inspired by methods for aligning phrases and words in texts of different languages developed for constructing dictionaries for machine translation. The results are competitive with state of the art performance on the "Labeled Faces in the Wild" dataset in terms of recall values, now reported on the complete dataset, include excellent precision values, and show the value of text and image analysis for identifying the probability of being pictured or named in the alignment process.

Keywords

Cross-media mining, image annotation.

1. THE CONTRIBUTIONS OF OUR WORK

In this paper, we address the challenge of aligning data across different modalities. In particular, we focus on the problem of aligning the names found in an image caption with the faces found in the corresponding image and vice versa. Such cross-media alignment brings a better understanding of the cross-media documents as it couples the different sources of information together and allows to resolve ambiguities that may arise from a single media document analysis (e.g. confusion between senior and junior George Bush). At the same time, it builds a cross-media model for each person in a fully unsupervised manner, which in turn allows to name the faces appearing in new images (with or without caption) or to show a picture of the people mentioned in new texts. Because there are usually several names mentioned in the text and several faces shown in the image (see figure 1), and not all of the names have a corresponding face and vice versa, there are many possible alignments to choose from, making cross-media linking a non-trivial problem. However, analyzing a large corpus of cross-media stories (images with captions) the re-occurrence over and over again of particular face-name pairs provides evidence that they might indeed be linked to the same person. This is based on the assumption that the two modalities are correlated at least to some extent - a reasonable assumption for news stories where both modalities give a description of the same event.

This problem has been studied before (e.g. [2, 1, 3, 5, 6]). However, in earlier work the stress has always been on assigning names to the faces in the images. Here, our aim is to broaden this, exploiting the (a)symmetry between the two modalities. In a first model, we assume that the names of the text generate the faces in the image; in a second model we assume that the faces in the image generate the names in the text, and in a third model we consider the joint probability of the names and faces in order to compute the alignment.

To implement the **first model**, one can think of the alignment of names and faces as the problem of assigning suitable faces to the names. For instance, given a text, the task could be to find a suitable illustration for it. In this case a text with names generates an image with faces. So, the task is to find a face f for a given name n. In each image-text pair s_i, given N_i names, there are many possible alignment schemes a_j to assign F_i faces and a null face to these names, from which we have to choose the best one. The constraint for each alignment scheme is that a face must be assigned only to one name, while the null face can be assigned to any name. When estimating the likelihood of an alignment scheme, the probability of a face given a name, $P(f|n)$, plays an important role.

*A full version of this paper is available in IEEE Transactions on Multimedia, Vol. 12, No. 1, January 2010.

Vice President **Dick Cheney** speaks at a luncheon for Republican U.S. Senate candidate **John Cornyn** Friday, July 19, 2002, in Houston. (AP Photo/Pat Sullivan)

President-elect **Barack Obama** is inching closer to naming former rival Sen. **Hillary Clinton** as his secretary of state, ABC News has learned. (Getty Images)

Danish director **Lars Von Trier** (C), Australian actress **Nicole Kidman** and Swedish actor **Stellan Skarsgard** (L) pose on a terrace of the Palais des festivals.(AFP/Boris Horvat)

Figure 1: Examples of stories each composed of an image and associated text.

To implement the **second model**, we can also inverse the above asymmetric assignment and assign names to the faces. For instance, given an image, the task could be to describe the image content with text. In this case an image with faces generates a text with names. So, the task is to assign a name n to a given face f. This is the usual way of looking at this problem, e.g. in the works of [2, 1, 3]. In each image-text pair s_i, given F_i faces, there are again many possible alignment schemes a_j to assign N_i names and a null name to these faces. The constraint for each alignment scheme is that a name must be assigned only to one face, while the null name can be assigned to any face. When estimating the likelihood of an alignment scheme, the probability of a name given a face, $P(n|f)$, plays an important role.

The **third model**, a stricter and more symmetric method, is to use the joint probability, $P(f, n)$, instead of the conditional probabilities $P(f|n)$ or $P(n|f)$. $P(f, n)$ represents the probability that a certain name and a certain face co-occur. This can be obtained by either using $P(n|f)$ or $P(f|n)$:

$$P(f, n) = P(f|n)P(n) = P(n|f)P(f) \quad (1)$$

It could be interpreted as follows. We no longer assume the names or faces to be given, but both are drawn from a random distribution in one of the following ways. In a data set (e.g., todays news) a name occurs with a certain prior probability and given this name, we pick a face with a certain probability; or when a prior probability of the occurrence of a face is known, we pick a name to describe it. The prior probability and how it is estimated has an important influence on the result, compared to the likelihood functions discussed in the previous sections. This could be considered as a Bayesian approach, where the two previous ones are more frequentist interpretations. In each image-text pair s_i, containing F_i faces and N_i names, there are again many possible alignment schemes a_j to combine them. In this setting, a null name can be assigned to any face except for the null face and a null face can be assigned to any name except for the null name.

We use here a standard Expectation Maximization algorithm, possibly augmented with a deterministic annealing component. Additionally, because not all names of the text are equally important and the same is true for the faces in the image, the models are corrected and improved based on the estimated salience information.

Our best results give a name and face alignment performance of 76.12% recall, and up to a 77.21% precision value, where the results are obtained based on the complete "Labeled Faces in the Wild" dataset [4].

The main contributions of our work include a more sophisticated face appearance model than used in [2], clustering of names and clustering of faces for finding correlations and initializing the EM algorithm, a modeling of the probability that a cited name is pictured or a shown face is named, based on salience, respectively called picturedness and namedness, a systematic and thorough analysis and evaluation of the alignment issues on the complete Faces in the Wild dataset. For picturedness computation we rely on natural language processing techniques such a noun phrase coreferent resolution and salience detection as used in text summarization, an approach which differs from [2]. We proved that integrating and combining picturedness and namedness in the likelihood functions improved the correct recognition of the alignments.

2. REFERENCES

[1] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 848–854, 2004.

[2] T. L. Berg, A. C. Berg, J. Edwards, and D. Forsyth. Who's in the picture. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 137–144. MIT Press, Cambridge, MA, 2005.

[3] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Automatic face naming with caption-based supervision. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[4] G. B. Huang, M. Rameh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[5] V. Jain and A. M. E. Learned-Miller. People-lda: Anchoring topics to people using face recognition. In *Proceedings International Conference on Computer Vision*, 2007.

[6] D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

Predicting the Volume of Comments on Online News Stories (Abstract)*

Manos Tsagkias e.tsagkias@uva.nl
Wouter Weerkamp w.weerkamp@uva.nl
Maarten de Rijke mdr@science.uva.nl

ISLA, University of Amsterdam
Science Park 107, 1098 XG Amsterdam

ABSTRACT

On-line news agents provide commenting facilities for readers to express their views with regard to news stories. The number of user supplied comments on a news article may be indicative of its importance or impact. We report on exploratory work that predicts the comment volume of news articles prior to publication using five feature sets. We address the prediction task as a two stage classification task: a binary classification identifies articles with the potential to receive comments, and a second binary classification receives the output from the first step to label articles "low" or "high" comment volume. The results show solid performance for the former task, while performance degrades for the latter.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics

General Terms

Algorithms, Theory, Experimentation, Measurement

Keywords

Comment volume, prediction, feature engineering

1. INTRODUCTION

As we increasingly live our life online, in the form of blogs, discussion forums, comment facilities, etc., new types of data become available that can be mined for valuable knowledge. E.g., online chatter can be used to predict sales ranks of books [4]. Online news is an especially interesting data type for mining and analysis purposes. Much of what goes on in social media is a response to news events, as is evidenced by the large amount of news-related queries users submit to blog search engines [9]. Tracking news events and their impact as reflected in social media has become an important activity of media analysts [1]. We focus on online news articles plus the comments they generate, and attempt to predict news article comment volume prior to publication time.

*The full version of this paper appeared in *CIKM 2009*.

DIR'10, January 25, 2010, Nijmegen, the Netherlands.

One might raise the question why one should be interested in commenting behavior and the factors contributing to it. We envisage three types of application for predicting the volume of comments generated by news articles. First, *media and reputation analysis* is dependent on what users think of topics covered in the media. Predicting the comment volume might help in determining the desirability of an article (e.g., regarding the influence on one's reputation) or the timing of its publication (e.g., generate publicity and discussion during election time). Second, *pricing of news articles* by news agencies and *ad placement strategies* by news publishers could be made dependent on the expected comment volume; articles that are more likely to generate comments could be priced differently. Finally, news consumers could be served only news articles that are most likely to generate many comments; news sources can thus provide new services to their customers and can *save consumers' time* in identifying "important" articles.

Our aim in this paper is to predict comment volume of news articles prior to publication. To this end, we seek to answer the following two questions: (i) What are the dynamics of user generated comments on news articles? We look at article and comment statistics per source. (ii) Can we predict, prior to publication, whether a news story will receive any comments at all, and if so, whether it will receive few or many comments?

This work makes several contributions. First, it explores the dynamics of user generated comments in on-line Dutch media. Second, it introduces the problem of predicting the comment volume of a news article. Third, it provides a set of surface, cumulative, textual, semantic, and real-world features that can be used to predict the number of comments of a news story prior to publication. Fourth, it provides an evaluation of the introduced features. Fifth, an error analysis identifies possible causes for classification failure.

Section 2 contains related work; we explore news comments in Section 3; our feature sets are introduced in Section 4; predicting comment volume is done in Section 5; Section 6 contains discussion, error analyses, conclusions, and future work.

2. RELATED WORK

Different aspects of the comment space dynamics have been explored in the past. Schuth et al. [11] explore the news comments space of four on-line Dutch media, while Mishne and Glance [10] explored the weblog comment space. Kaltenbrunner et al. [6] measured community response time in terms of comment activity on Slashdot stories, and discovered regular temporal patterns on people's commenting behaviour. Recently, various prediction tasks and correlation studies have been considered in social media. Mishne and de Rijke [8] use textual features as well as temporal metadata of blog posts to predict the mood of the blogosphere. De Choudhury et al. [3] correlate blog dynamics with stock market activity,

and Gruhl et al. [4] perform a similar task with blogs/reviews and book sales. Szabó and Huberman [12] predict the popularity of a story or a video on Digg or YouTube, given an item's statistics over a certain time period after publication. Lerman et al. [7] forecast the public opinion of political candidates from objective news articles. Finally, Tsagkias et al. [13] predict podcast preference using surface features extracted from podcast RSS feeds.

To our knowledge, no prediction tasks have been published that concern the volume of comments generated by online news articles.

3. EXPLORING NEWS COMMENTS

Our data consists of the aggregated content from seven on-line news agents: *Algemeen Dagblad* (*AD*), *De Pers*, *Financieel Dagblad* (*FD*), *Spits*, *Telegraaf*, *Trouw*, *WaarMaarRaar* (*WMR*), and one collaborative news platform, *NUjij*. We have chosen to include sources that provide commenting facilities for news stories, but differ in coverage (regional/national), in political views, in subject (general/politics/arts/entertainment), and in type.

We turn to our first research question: What are the dynamics of user generated comments on news articles? Our data exploration reveals "big" and "small" news agents based on their respective number of published articles and received comments. User commenting behaviour is measured twofold: a) as the reaction time required for an article to receive a comment, and b) as the discussion timespan, for how long an article keeps receiving comments. Although we recorded variations of commenting behaviour between sources yet our findings are comparable to commenting behaviour in blogs [10]. These aspects of online news seem to be inherent characteristics of each source, possibly reflecting the credibility of the news organization, the interactive features they provide on their web sites, and their readers' demographics [2]. Our features attempt to capture the differences between the sources into account.

4. FEATURE ENGINEERING

We consider five groups of features: a) *surface*: captures feed metadata quality, b) *cumulative*: identifies impact of a news article by monitoring how many times an article's near-duplicate appears in our dataset, c) *textual*: captures which terms are correlated with most and least commented articles, d) *semantic*: similar to *textual* captures discriminative entities and locality, and e) *real-world*: outside temperature at time of publication.

5. PREDICTING COMMENT VOLUME

We now turn to the second research question: Can we predict, prior to publication, whether a news story will receive any comments at all? And if it receives comments, can we predict whether it receives few or many comments?

We address the prediction task as two consecutive classification tasks to compensate for the highly skewed datasets. First, we segregate articles with regard to their potential of receiving comments. A binary classification is performed with two classes: *with comments* vs. *without comments*. Second, we predict the comment volume level for the articles predicted to receive comments in the first step (positive class). This second classification is performed with two classes: *low volume* and *high volume*.

For the first classification step our results show high F1-scores for most sources but with low Kappa-statistic. Textual and semantic features perform the best among all feature sets. The run with all features combined did not lead to substantial improvements over the individual features.

For the second classification step F1-scores drop compared to previously. Textual and semantic features are again strong performers, although they exhibit high variance over the board. All features combined lead to better performance over the baseline. The lower Kappa-values of this run indicate more robust classification.

6. DISCUSSION AND OUTLOOK

We presented exploratory work on predicting the comment volume of news articles prior to publication. We have developed a set of surface, cumulative, textual, semantic, and real-world features and report on their individual and combined performance on two classification tasks: Classify articles according to whether they will (i) generate comments, and (ii) receive few or many comments. Our experiments show that predicting the volume of comments is more difficult than predicting whether an article will receive any comments at all. Textual and semantic features prove to be strong performers, and the combination of all features leads renders classification more robust. Our failure analysis indicates that the features used in this paper are not the only factors involved in the prediction process. Future work should therefore focus on extracting more feature sets (e.g., context and entity-relations), use different encodings for current features, optimize the number of textual and semantic features per source, and explore optimized feature sets.

Acknowledgments. This research was supported by the DuOMAn project (STE-09-12) carried out within the STEVIN programme and by the Netherlands Organisation for Scientific Research (NWO) under project numbers 017.001.190, 640.001.501, 640.002.501, 612.066.512, 612.061.814, 612.061.815, 640.004.802.

7. REFERENCES

[1] D. L. Altheide. *Qualitative Media Analysis (Qualitative Research Methods)*. Sage Pubn Inc, 1996.

[2] D. S. Chung. Interactive features of online newspapers: Identifying patterns and predicting use of engaged readers. *J. Computer-Mediated Communication*, 13(3):658–679, 2008.

[3] M. De Choudhury, H. Sundaram, A. John, and D. D. Seligmann. Can blog communication dynamics be correlated with stock market activity? In *HT '08*, pages 55–60. 2008.

[4] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *KDD '05*. 2005.

[5] A. Kaltenbrunner, V. Gomez, and V. Lopez. Description and prediction of slashdot activity. In *LA-WEB '07*, pages 57–66. IEEE Computer Society, 2007.

[6] A. Kaltenbrunner et al. Homogeneous temporal activity patterns in a large online communication space. *CoRR*, abs/0708.1579, 2007.

[7] K. Lerman, A. Gilder, M. Dredze, and F. Pereira. Reading the markets: Forecasting public opinion of political candidates by news analysis. In *Coling 2008*, pages 473–480, 2008.

[8] G. Mishne and M. de Rijke. Capturing global mood levels using blog posts. In *AAAI-CAAW '06*, pages 145–152, 2006.

[9] G. Mishne and M. de Rijke. A study of blog search. In *ECIR '06*, pages 289–301, 2006.

[10] G. Mishne and N. Glance. Leave a reply: An analysis of weblog comments. In *WWE '06*, 2006.

[11] A. Schuth, M. Marx, and M. de Rijke. Extracting the discussion structure in comments on news-articles. In *WIDM '07*, pages 97–104. ACM, 2007.

[12] G. Szabó and B. A. Huberman. Predicting the popularity of online content. *CoRR*, abs/0811.0405, 2008.

[13] E. Tsagkias, M. Larson, and M. de Rijke. Exploiting surface features for the prediction of podcast preference. In *ECIR '09*, 2009.

Combining Query by Navigation with Query by Example

Ferdi van der Werf
Institute for Computing and Information Sciences
Radboud University of Nijmegen
Nijmegen, The Netherlands
efcm@vdwerf.eu

Franc Grootjen
Donders Institute of Brain, Cognition and Behaviour
Centre of Cognition
Radboud University of Nijmegen
Nijmegen, The Netherlands
grootjen@acm.org

Louis Vuurpijl
Donders Institute of Brain, Cognition and Behaviour
Centre of Cognition
Radboud University of Nijmegen
Nijmegen, The Netherlands
l.vuurpijl@donders.ru.nl

ABSTRACT

Despite the efforts to reduce the semantic gap between user perception of similarity and feature based representation of images, user interaction seems to be essential to improve retrieval performances in content based image retrieval. The two classic ways of dealing with this problem are improving feature extraction/selection and exploiting relevance feedback.

This paper suggests an alternative solution. It proposes a model in which the richness of textual conceptual representations are mapped to traditional Query by Example techniques. By doing so the system is able to benefit from the advantages of both worlds.

The model will be illustrated with an example run on a real world database of annotated images.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content analysis and indexing

Keywords

Formal Concept Analysis, Content Based Image Retrieval, Query by Navigation, Query by Example, Conceptual Knowledge Representation.

1. INTRODUCTION

Given their simple nature, the success of text or keyword-based retrieval systems is astonishing. Although these methods apparently only process words (and their word counts), they rely on and endorse most of their success to the keywords' *implicit semantics*. In fact each keyword is a representation of a thought, or *concept*.

Both this success as well as the ease of textual query formulation are the main reason for modern image retrieval systems to use keyword based systems. Especially when the image data is accompanied by (descriptive) text like web pages, or when images are annotated manually.

During the last decade the field of Content Based Image Retrieval made huge steps in object recognition, feature and texture extraction. These advancements however, seems somehow to have stalled. At least two problems are hindering the progress:

1. There exists a gap (i.e. *semantic gap*) between the high level semantic meanings of images and their low-level features [14].

2. Apart from being low-level, the features are hard to describe or conceptualized. This problem becomes apparent when trying to formulate a query.

In current research the problem of the semantic gap is mainly attacked by technical improvements in feature selection and selection methods [13] or exploiting relevance feedback [5, 14]. Others try to bridge the gap by using methods that are inspired by humans' visual perception [27]. In the field of video retrieval more progress has been reported, mainly because of the richness of the platform. A recent overview of concept based video retrieval is presented in [24].

The representational issues of the features are mainly countered by *Query by Example*. This method, suggested by Luhn [15] and coined by [29] for well structured relational databases, provides a workaround by using one or more example images as query. Tightly coupled with relevance feedback, the GIFT system [16] is build around this searching paradigm. Another approach worth mentioning is the outline description method [23] in which the searcher can manually draw the outline of an item of interest which is used as a query.

In this research we will use counter problems sketched above by using the conceptual richness of keywords by creating a conceptual structure using Formal Concept Analysis [28]. This structure has nice navigational properties which support *Query by Navigation*. Each conceptual node in the structure will be mapped to images using Query by Example and, by doing so we will show that the resulting system has the benefits of both worlds.

2. THE MODEL

In order to automatically combine image with text retrieval, we will need a model that somehow captures the 'meaning' of terms in documents. We will show that by using Formal Concept Analysis, introduced by Wille [28] and Hardegree [9], we can create a mathematical structure which can be used to semantically classify documents in formal concepts. These structures (called lattices) have nice mathematical properties and are a starting point for navigational systems [3].

2.1 Concept Lattices

Mathematically seen concept lattices are the result of a Galois connection, coined by [18]. Their value for Information Retrieval was already apparent to Salton who described aspects of concept lattices in the first edition of [21][1]. As shown in [7] concept lattices are a special form of Dualistic Ontologies. More recent research uses Formal Concept Analysis to describe the complex notion of Information Need [10].

2.1.1 Context

We denote the collection of documents with the letter $\mathcal{D}$. Individual members of this collection (documents) are written with small letters like d, d_1, d_2, ..., while subsets are written in capitals D, D_1, D_2, ... During the indexing process, descriptors (attributes) are attached to the documents. We write $\mathcal{A}$ to denote the set of all attributes, a, a_1, a_2, ...for individual attributes, and A, A_1, A_2, ...for attribute sets (subsets of $\mathcal{A}$). The result of the indexing process is reflected by the binary relation $\sim$: we write $a \sim d$ iff attribute a describes document d. The tuple $(\mathcal{D}, \mathcal{A}, \sim)$ is called a *context*. To simplify the notation we will overload the context relation $\sim$ to cover set arguments:

$$a \sim D \stackrel{\text{def}}{\equiv} \underset{d \in D}{\forall} [a \sim d]$$

$$A \sim d \stackrel{\text{def}}{\equiv} \underset{a \in A}{\forall} [a \sim d]$$

2.1.2 Example context

Imagine we have a collection of 8 documents, $\mathcal{D} = \{d_1, d_2, \ldots, d_8\}$. The documents are indexed with their keywords, $\mathcal{A} = \{$*Cab, Beach, Airplane, Tourist, Car, Loading, Hut, Holiday*$\}$. The corresponding context relation is depicted in table 1, see figure 1 for their images.

	Cab	Bea	Air	Tou	Car	Loa	Hut	Hol
d_1			x					
d_2	x							
d_3					x			
d_4		x		x				
d_5			x			x		
d_6		x					x	
d_7		x						
d_8		x						x

Table 1: Example context relation $\sim$

[1]For some reason these aspects did not make it to the second edition.

2.1.3 Context properties

Using the context relation a classification of documents and attributes can be generated such that each class can be seen as a concept in terms of properties of the associated documents and attributes. In our interpretation, documents and attributes assign meaning to each other via the context relation: within the limits of this view we can not distinguish between documents with identical properties, while attributes having the same extensionality are assumed to be identical. Sharing *document meaning* thus can be seen as sharing attributes:

Definition 1
The common attributes of a set of documents are found by the right polar function **ComAttr**: $\mathcal{P}(\mathcal{D}) \rightarrow \mathcal{P}(\mathcal{A})$ defined as follows:

$$\mathbf{ComAttr}(D) = \{a \in \mathcal{A} \mid a \sim D\}$$

Of course, documents may also be shared by attributes:

Definition 2
The documents sharing properties are captured by the left polar function
ComDocs: $\mathcal{P}(\mathcal{A}) \rightarrow \mathcal{P}(\mathcal{D})$ defined by:

$$\mathbf{ComDocs}(A) = \{d \in \mathcal{D} \mid A \sim d\}$$

Generally speaking, the right and left polar functions are not each other's inverse. However, a combination of a document set (and the corresponding attribute set) for which the polar functions behave as inverse are considered special: they both ascribe their meaning to each other. Such a combination is referred to as a *concept*:

Definition 3
A *concept* is a pair $(D, A) \in \mathcal{P}(\mathcal{D}) \times \mathcal{P}(\mathcal{A})$ such that D and A are their mutual meaning:

$$\begin{aligned} \mathbf{ComAttr}(D) &= A \\ \mathbf{ComDocs}(A) &= D \end{aligned}$$

Obviously not *every* set of documents (attributes) forms a concept. But when it does, at most one concept can be associated with it. So a concept is uniquely identified by its set of documents or by its set of attributes.

2.1.4 The concept lattice

Let $\mathcal{C}$ be the set of all concepts belonging to a context $(\mathcal{D}, \mathcal{A}, \sim)$. First we define an order on $\mathcal{C}$:

Definition 4
Let $c_1 = (D_1, A_1)$ and $c_2 = (D_2, A_2)$ be concepts, then

$$c_1 \subseteq c_2 \stackrel{\text{def}}{\equiv} D_1 \subseteq D_2$$

The duality of documents and attributes is nicely exposed by the following lemma:

Lemma 1
Let $c_1 = (D_1, A_1)$ and $c_2 = (D_2, A_2)$ be concepts, then

$$D_1 \subseteq D_2 \Longleftrightarrow A_1 \supseteq A_2$$

d_1 d_2 d_3 d_4

d_5 d_6 d_7 d_8

Figure 1: Example documents $d_1 - d_8$

The partial order on concepts forms a *lattice*. It has a bottom element, corresponding with overspecification, and a top which represents underspecification.

2.2 Concept Lattice generation

There are several ways to calculate the concepts belonging to a given context. The most straightforward (brute force) way is just testing every subset of $\mathcal{D}$. Of course this method is not feasible for non-trivial set sizes. A more practical way is to generate the lattice bottom-up: starting with a set of base concepts and using the lattice property that each set of concepts has a smallest upperbound. Finding base concepts is easy using the following lemma:

Lemma 2
Let $D \in \mathcal{D}$, then

$$(\mathbf{ComDocs}(\mathbf{ComAttr}(D)), \mathbf{ComAttr}(D)) \text{ is a concept}$$

Applying lemma 2 to all singleton subsets of $\mathcal{D}$ yields all base concepts.

More elaborated algorithms like described in [4] may be used to handle very big contexts. If execution time is important even parallel implementations like [2] may be considered. Figure 2 shows the Hasse diagram of the example context.

2.3 Query by Navigation

Since the concepts in a lattice are *ordered* they facilitate a form of navigation. This navigation process involves moving the focus from concept to concept, guided by the difference between the focus and target nodes. The resulting mechanism is called *Query by Navigation* [3] and is especially useful in situations where the information need is unclear or incomplete. In this section we will demonstrate the use of traditional lattice navigation [6, 20].

Each node in the lattice represents a conceptual unit, described by the combination of their *extension* (the documents) and their *intention* (keywords or features). Navigation up means losing specificness; i.e., decreasing the number of attributes and increasing the number of documents. Navigating down represents getting more specific; i.e., increasing the number of attributes and decreasing the number of documents.

Assume we start the navigation in the top node of the keyword concept lattice (Figure 2). This node represents all documents. Possible navigation steps are:

- Add keyword 'beach'
- Add keyword 'car'
- Add keyword 'airplane'
- Add keyword 'cab'

We will navigate down left to the concept labeled with the 'beach' attribute. The extension of this concept is $\{d_4, d_6, d_7, d_8\}$; all document picturing a beach scene. From this concept the following navigation steps are possible:

- Drop keyword 'beach'
- Add keyword 'holiday'
- Add keyword 'hut'
- Add keyword 'tourist'

Obviously, these navigation steps will bring us back to the top concept or to the corresponding concept containing a single document d_4, d_6 or d_8.

2.4 Query by Example

During the navigational process searchers may end up in node that textually describes their need. However, since only textual information is used to generate the concept, there may be more images in the collection that are relevant. This is where the (traditional) field of Content Based Image Retrieval comes in. Because every concept consists of meaningfully coupled attributes (keywords) *and* documents (images) we are able to generate a query for each concept. Note that this way there are no representational issues with

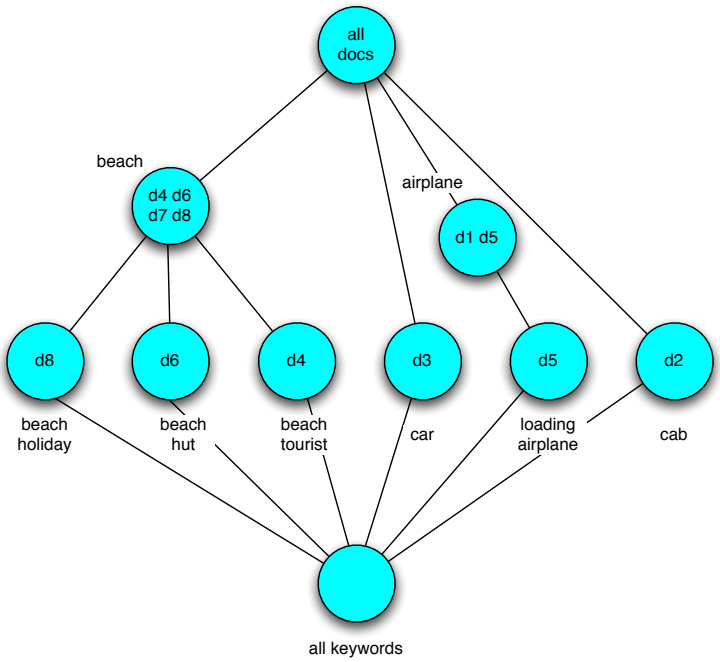


Figure 2: Example lattice

features, since we simply use the image(s) found in the concept and use *Query by Example*. This way the searcher benefits from the descriptive and conceptual representation of the keyword based textual system, without losing the advantages that Content Based Image Retrieval systems offer by calculating image similarities.

3. THE EXPERIMENT

In order to test our model, we decided to run an experiment on real world data. We selected the IAPR TC-12 collection [8] since it contains a realistic number of images accompanied by (English) textual descriptions.

3.1 Lattice generation

The collection consists of 20,000 still natural images taken from locations around the world and comprising an assorted cross-section of still natural images. This includes pictures of different sports and actions, photographs of people, animals, cities, landscapes and many other aspects of contemporary life.

The English descriptions contained 12630 unique words, 10500 unique words after stopword removal and lowercase conversion, and 8800 unique words after porter stemming[19].

For the experiment we selected 100 keywords (to be more precise their lemmas or stems) and created a context which combines these keywords with the 20,000 images. We used Yanacona [2] to generate the corresponding concept lattice which contained 15,228 concepts.

3.2 Content Bases Image Retrieval

To analyze the images and extract their features we used the open-source Content Based Image Retrieval System GIFT (the GNU Image-Finding Tool). This system is designed around the *Query By Example* paradigm: it searches for images using (a set) of example images. GIFT has been developed by The Computer Vision Group of the University of Geneva [1]. The first version of GIFT, known as Viper [26], was released in 1999. The current version of GIFT is 0.1.14 and is still under development.

To facilitate third party research and plugin development for GIFT, the authors designed MRML (Multimedia Retrieval Markup Language). MRML [25] is a XML based language that offers a clean separation between the client and the server of the retrieval system. It describes a standard method for communication with other CBIR systems [17]. In order to connect to the GIFT server we developed navigational software that communicates with the server using MRML.

GIFT employs more than 80.000 simple color and spatial frequency features, both local and global, extracted at several scales. These are intended to correspond (roughly) to features present in the retina and early visual cortex [26].

3.2.1 Features

According to [26] Gift uses a palette of 166 colors, derived by quantizing HSV space into 18 hues, 3 saturations, 3 values and 4 gray levels. Two sets of features are extracted from each image. The first is a color histogram, with empty bins discarded. The second represents color layout. Each block in the image (starting with the image itself) is recursively divided into four, at four scales. The mode color of each block is treated as a binary feature, meaning that there are 56 440 possible color block features.

Gabors have been applied to texture characterization, as well as more general vision tasks.

In order to produce a ranked listing of images GIFT uses classical information retrieval techniques like *tfidf* [11, 22] and inverted files[12].

3.3 Example run

In order to test the system we will perform an example run on our data. The run consists of two phases, a Query by Navigation phase followed by a Query by Example phase.

3.3.1 Query by Navigation

Using an adapted version of NavCon [20] we navigate through the concept lattice. This tool enables us to navigate our focus up or down the lattice by adding or dropping keywords. After each navigation step clickable thumbnails are presented of the images located at the current focus. Notice that since the presented images are members of the current concept, they all share the same keywords. Because of this, they are the direct result of the textual disclosure. After navigating with the keywords 'sunset' and 'sky' we end up in a lattice node containing the thumbnails shown in figure 3.

3.3.2 Query by Example

After hitting the QBE button the system will use the images belonging to the current focus to do a Query by Example retrieval run of GIFT. The ranked result (top 5) images will be added to the result screen. Note that this step does not involve any textual operations: it may well be the case that none of the selected keywords are present in the descriptions of these new images. See figure 4 for the resulting images after the Query by Example run by GIFT.

4. CONCLUSION AND FUTURE WORK

We have presented a model that is able to represent both textual conceptual information as image features of annotated images. We showed that it is possible to disclose a real world collection of images using Query by Navigation and Query by Example.

Future research might investigate the effectiveness of this disclosure in several ways:

- By using a collection which is accompanied by textual queries and relevance judgments the effectiveness of the system could be measured.
- Starting with a fully annotated collection one might establish a baseline for a set of queries. Subsequently 50% of images could be stripped from their description. Operating on this crippled collection the system's performance gained by the Query by Example part could be tested.

This research will result in a web accessible demonstrator at `http://vindit.ai.ru.nl` available mid-2010.

5. ACKNOWLEDGMENT

This research is carried out within the ToKeN VindIT project (grant number 634.000.018) of the Netherlands Organization for Scientific Research (NWO).

6. REFERENCES

[1] The Computer Vision Group of the University of Geneva. `http://vision.unige.ch`.

[2] M. Blokpoel, F. A. Grootjen, and E. L. van den Broek. Exploiting coarse grained parallelism in conceptual data mining. In *Proceedings of the Dutch Information Retrieval Conference (DIR2008)*, Maastricht, April 2008.

[3] P. D. Bruza and T. P. van der Weide. The modelling and retrieval of documents using index expressions. *SIGIR Forum*, 25(2):91–103, 1991.

[4] B. Ganter. Two basic algorithms in concept analysis. Technical Report FB4-Preprint No. 831, TH Darmstadt, 1984.

[5] G. Giacinto, F. Roli, and G. Fumera. Adaptive query shifting for content-based image retrieval. In *MLDM '01: Proceedings of the Second International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 337–346, London, UK, 2001. Springer-Verlag.

[6] F. A. Grootjen. A semantical twist to syntactical navigation. In *Proceedings of the 2^{nd} international workshop on logical and uncertainty models for information systems (LUMIS2000)*, Greenwich, London, England, September 2000.

[7] Grootjen, F.A. and van der Weide, Th. P. Dualistic ontologies. *International Journal of Intelligent Information Technologies*, 1(3):1–20, July 2005.

[8] M. Grubinger, P. D. Clough, H. Müller, and T. Deselaers. The IAPR benchmark: A new evaluation resource for visual information systems. In *International Conference on Language Resources and Evaluation*, Genoa, Italy, 24/05/2006 2006.

[9] G. M. Hardegree. An approach to the logic of natural kinds. In *Pacific Philosophical Quarterly*, volume 63, pages 122–132, 1982.

[10] E. Hoenkamp. On the notion of "an information need". In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pages 354–357, Berlin, Heidelberg, 2009. Springer-Verlag.

[11] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[12] D. E. Knuth. *The art of computer programming / Donald E. Knuth.* Addison-Wiley, Reading, Mass., :, 1968.

[13] C.-H. Lin, R.-T. Chen, and Y.-K. Chan. A smart content-based image retrieval system based on color and texture feature. *Image Vision Comput.*, 27(6):658–665, 2009.

[14] Y. Liu, X. Chen, C. Zhang, and A. Sprague. Semantic clustering for region-based image retrieval. *J. Vis. Comun. Image Represent.*, 20(2):157–166, 2009.

[15] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.*, 1(4):309–317, 1957.

[16] H. Mueller, D. M. Squire, W. Mueller, and T. Pun. Efficient access methods for content-based image retrieval with inverted files. volume 3846, pages 461–472. SPIE, 1999.

[17] H. Müller, W. Müller, D. M. Squire, Z. Pečenović,

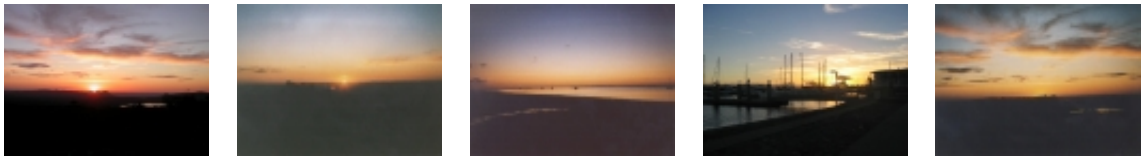


Figure 3: Images found in concept after Query by Navigation steps 'sunset' and 'sky'

Figure 4: Extra images found after Query by Example using GIFT

S. Marchand-Maillet, and T. Pun. An open framework for distributed multimedia retrieval. In *Recherche d'Informations Assistée par Ordinateur (RIAO'2000) Computer-Assisted Information Retrieval*, volume 1, pages 701–712., Paris, France, apr 12-14 2000.

[18] O. Ore. Galois connexions. *Transactions of the American Mathematical Society*, 55(3):493–513, 1944.

[19] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[20] W. Roelofs and F. A. Grootjen. Navcon, Navigating the conceptual space. In M. M. Dastani and E. de Jong, editors, *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence (BNAIC2007)*, pages 447–448, Utrecht, the Netherlands, November 2007.

[21] G. Salton. *Automatic Information Organization and Retrieval*. McGraw Hill Text, 1968.

[22] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[23] L. Schomaker, E. d. Leau, and L. Vuurpijl. Using pen-based outlines for object-based annotation and image-based queries. In *VISUAL '99: Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 585–592, London, UK, 1999. Springer-Verlag.

[24] C. G. M. Snoek and M. Worring. Concept-based video retrieval. *Found. Trends Inf. Retr.*, 2(4):215–322, 2009.

[25] D. M. Squire. Multimedia retrieval markup language. `http://www.mrml.net`.

[26] D. M. Squire, W. Müller, H. Müller, and T. Pun. Content-based query of image databases: inspirations from text retrieval. *Pattern Recogn. Lett.*, 21(13-14):1193–1198, 2000.

[27] E. L. van den Broek, T. E. Schouten, and P. M. F. Kisters. Modeling human color categorization. *Pattern Recogn. Lett.*, 29(8):1136–1144, 2008.

[28] R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, *Ordered sets*, pages 445–470. D. Reidel Publishing Company, Dordrecht–Boston, 1982.

[29] M. M. Zloof. Query-by-example: the invocation and definition of tables and forms. In *VLDB '75: Proceedings of the 1st International Conference on Very Large Data Bases*, pages 1–24, New York, NY, USA, 1975. ACM.

Amharic Question Answering (AQA)

Seid Muhie Yimam
Department of Information Technologies
Adama University
Adama, Ethiopia
+251913604368
seidyimam@yahoo.com,
seidymam@gmail.com

Mulugeta Libsie
Department of computer Science
Addis Ababa University
Addis Ababa, Ethiopia
mlibsie@cs.aau.edu.et, mlibsie@yahoo.com

ABSTRACT

The number of Amharic documents on the Web is increasing as many newspaper publishers started providing their services electronically. People were relying on IR systems to satisfy their information needs but it has been criticized for lack of delivering "readymade" information to the user so that Question Answering systems emerge as the best solution to get the required information to the user with the help of information extraction techniques. In this paper we have developed a Question Answering System for Amharic (AQA). The language specific issues in Amharic are extensively studied and hence, document normalization was found very crucial for the performance of our Question Answering system. Novel technique were developed to determine the question types, possible question focuses, and expected answer types as well as to generate proper Information Retrieval query, based on our language specific issue investigations. An approach in document retrieval focuses on retrieving three types of documents (Sentence, paragraph, and file). An algorithm has been developed for sentence/paragraph re-ranking and answer selection. The named-entity-(gazetteer) and pattern-based answer pinpointing algorithms developed help locating possible answer particles in a document. The rule-based question classification module classifies about 89% of the question correctly. The document retrieval component shows greater coverage of relevant document retrieval (97%) while the sentence based retrieval has the least (93%) which contributes to the better recall of our system. The gazetteer-based answer selection using a paragraph answer selection technique answers 72% of the questions correctly which can be considered as promising. The file based answer selection technique exhibits better recall (91%) which indicates that most relevant documents which are thought to have the correct answer are returned.

KEYWORDS: Amharic Question Answering, Answer Selection Techniques, Sentence/paragraph Re-ranking, Question Answering Evaluation

1. INTRODUCTION

The traditional information retrieval techniques were considered insufficient in retrieving precise information to the user. While information retrieval is effective by itself, users these days demand a better tool. First, they want to reduce the time and effort involved in formulating effective queries for search engines (users are required to formulate queries that should maximize document matching and the search engine processes the query as submitted), and secondly they want their results to be real answers - not the list of relevant links. Automatic question answering has become an interesting research area and has resulted in substantial improvement in its performance [1]. The aim of question answering (QA) is to retrieve exact information from a large collection of documents such as those on the Web. The main initiative behind QA systems development is that users in general prefer to have a single answer or a couple of answers for their questions rather than having a number of documents to be read as it happens with the output of search engines [2]. Having a huge number of documents such as the World Wide Web or a local collection, a QA system should be able to retrieve answers to questions formulated in natural language. QA systems have already been developed in different languages such as Chinese [3, 4, 5], English [6, 7] and so on.

This research is about Amharic Question Answering (AQA) System (**ጠጠየቅ**), which is the first of its kind. This is because the number of Amharic documents on the Web is increasing gradually as many newspaper agencies started to provide their services electronically. Our QA system has been given the name **ጠጠየቅ** *(Be questioned),* a historical verbalism in Ethiopia where two people appear before a judge used to ask a question for the defendant which are of kind ironic**.** Amharic is written with a version of the Ge'ez script known as **ፊደል** (Fidel). The Amharic language has its specific way of grammatical construction, character (fidel) representation, and statement formation [8, 9, 10] where a question answering system depends on all for question processing and answer selection techniques.

The question construction and answering techniques in Amharic language are different from English and other languages. In English, questions will be developed, for example, using "***wh***" words such as "Who is the Prime Minister of Ethiopia?". But this same question will have a different structure in Amharic such as a difference in character and word formation as well as grammatical arrangement and type of question particles (terms used to ask questions) used. For example, the above question will be translated to (**የኢትዮጵያ ጠቅላይ ሚኒስትር ማን ይባላሉ? -** *Yeethiopia Teqlay minister man yibalalu*). This question needs a special consideration to exactly return the correct answer, which is very different from English and other languages question answering techniques. In this paper, we investigated the problems and limitations of an Amharic search engine, the effect of developing a QA system, analyze

the strengths and weaknesses of QA with respect to search engines and developed an Amharic question answering system.

2. QUESTION ANSWERING SYSTEMS

Information Retrieval (IR) has been researched extensively mainly to help users in getting relevant documents from large collection of free-text documents. The way IR tackles the problem of document retrieval is based on the closeness of the document and the query submitted to the IR system. IR will not try to present answers to users explicitly. This was the critics of IR so that the need of Information Extraction (IE) came about. The IE technique involves NLP tools for precisely indicating a correct text. There should be deep analysis of queries (i.e., user questions) to understand the user's intention as well as deep analysis of the document to extract correct answers (sentences or passages). In the case of IR, a simple technique is sufficient to extract content-rich words from the query and applying stemming to make more uniformity of document retrieval that will be applied during indexing too [15].

A typical pipeline question answering architecture has four components; question analysis, document retrieval, passage (sentence) retrieval and answer extraction [13,14]. In this architecture, the Question Analyzer is responsible to analyze the question that is determining the proper expected answer type and formulating proper queries for the Document Retriever. The Document Retriever will retrieve the top n related documents that will be subjected to the Passage Retriever later. The Passage Retriever will extract passages that pinpoint possible answer strings. The final component, Answer Extractor, will extract the correct answer from the ranked extracted passages.

3. RELATED WORK

There are many related works in the literature. Here, we will present some of them that are relevant to our work. The work in [16] investigated a number of techniques for open-domain question answering. Investigated techniques include: manually and automatically constructed question analyzers, document retrieval specifically for question answering, semantic type answer extraction, and answer extraction via automatically acquired surface matching text patterns. Besides Factoid QA techniques, the paper briefly investigated approaches in definitional questions. The novel techniques in the paper are combined to create two end-to-end question answering systems which allow answers to be found quickly. The first is AnswerFinder[1] which answers factoid questions such as *"When was Mozart born?"*, whilst the second, Varro, builds definitions for terms such as "what is *aspirin?"*, "what does *Aaron Copland* mean?", and "define *golden parachute"*. Both systems allow users to find answers to their questions using web documents retrieved by Google. Together, these two systems demonstrated that the techniques developed can be successfully used to provide quick and effective open-domain question answering.

[1] *AnswerFinder* is a project that is developed by the Centre for Language Technology at Macquarie University

Marsha Chinese question answering system [18] focuses on evaluating techniques employed for the Chinese language. Marsha consists of three components, query processing, information retrieval (Hanquery search engine), and answer extraction just like many other QA systems. The query processing component analyzes the Chinese questions submitted and formulates a formal query that will be submitted to the IR component. The search engine component retrieves related candidate documents from the database. The answer extraction module extracts correct answers or candidate answers that are presented to the user.

The paper in [19] focuses on developing Hindi QA system which has different language constructs, query structure, common words, and so on as compared to English. The main aim of the paper was to help elementary and high school students in getting correct answers for their subject related questions whereby facilitating e-learning in the Hindi language. For query construction, the researchers used self constructed lexical database of synonyms since there was no Hindi WordNet available. A case-based rule has been developed to classify questions. After the user question is changed to a proper query (by applying stop-word deletion, domain knowledge entity inclusion, and so on), the query will be submitted to the retrieval engine. Finally answer selection is done by extensive analysis of passages and the correct answer will be presented to the user.

4. THE AMHARIC LANGUAGE

Amharic is a Semitic language spoken in many parts of Ethiopia. It is the official working language of the Federal Democratic Republic of Ethiopia and thus has official status nationwide. It is also the official or working language of several of the states/regions within the federal system. Outside Ethiopia, Amharic is the language of millions of emigrants (notably in Egypt, Israel and Sweden), and is spoken in Eritrea [11]. It is written using a writing system called *fidel* or *abugida*, adapted from the one used for the now-extinct Ge'ez language. Ethiopic characters (fidels) have more than 380 Unicode representations (U+1200-U+137F) [12].

In every language, questions are constructed with the help of question particles (interrogative words) and question marks (?) which is placed at the end of the question. Amharic also has its own question particles some of which are shown in Table 1 [20].

Table 1. Some Amharic Question Particles

Question word	Transliteration	Description
ማን	*man*	Who …
ለማን	*leman*	to whom
ማነው	*manew*	Who is …
የት	*yet*	Where …
ስንት	*sint*	How many
ለምን	*lemin*	Why …

There are different challenges in Amharic question answering. One of the main problems is that question particles by themselves can't help in determining the question type. Extra analysis is required to determine the question type so as to know the expected answer types. Secondly, some

proper names belong to more than one word categories, such as verb and noun so that determining whether that word is the expected proper name or not is very difficult. This problem is aggravated as there is no proper name capitalization in Amharic. Lastly, statement demarcation is a problem as there is no standardized writing by different writers.

5. DESIGN OF AQA

Every question answering system will have basic components of Question Analysis, Document retrieval and Answer Extraction [13, 14]. Our QA system has mainly five components, document pre-processing, question processing, document retrieval, sentence/paragraph re-ranking, and answer selection modules as shown in figure 1.

In the **document preprocessing** module, documents will be normalized to show similar standards for document retrieval and answer selection processing. Amharic is too specific in having different character representation with the same reading and writing style. For example, the character ሀ (*ha*) has five equivalent characters with the same reading style (same pronunciations with different spellings) and are used interchangeably. These are ሃ, ሐ, ሓ, ኀ, and ኃ. All occurrences of these characters should be replaced with ሀ (*ha*). The research shows that document processing improves the performance of our system nearly by 12 percent.

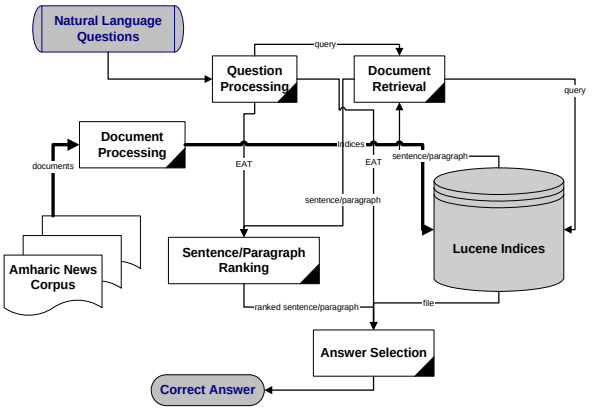


Figure 1. Design of AQA

Besides character normalization, we also did number normalization. Numbers in Amharic can be represented in Arabic numerals, Ethiopic numerals, or alphabetically using words. Number normalization helps to detect all possible numeric answer particles (expected answers, candidate answer strings) in the document which otherwise are left unmatched. Once character and number normalizations are done, stopwords are removed and proper stemming is performed. A total of 83 terms and all question particles are considered stop words. To delimit documents into sentence and paragraph, we have used different techniques. First sentences are detected with the Amharic full stop (።) if the document uses this punctuation mark. If the document uses group of Amharic word spaces (፡) or group of colons, we replace it with Amharic full stop. If the document is prepared with none of the punctuation marks mentioned, sentence finishing words such as ነው-*newu*, ታውቋል-*tawuquwal*, ተባለ-*tebale*, ገልጿል-*geltsuwal*, etc. are used. Similarly paragraphs are detected by the normal paragraph separator (new line followed by a blank line) or by an average number of sentences that can make up a paragraph. Once the document is normalized, sentences and paragraphs are delimited; then, the final task is to create sentence, paragraph and file indexes.

The **question processing** module accepts the user's question and performs tasks such as question type determination, question focus (important terms about the question) identification, and expected answer type determination. The question type will be determined based on the question particles and the question focuses. Since most of the question particles in Amharic are multipurpose, the question focus plays the greater role in determining the question type. We have developed a question typology that is be used to determine the expected answer type. The question processing module also generates the proper IR query that is submitted to the document retrieval component.

The question processing sub-component is shown in figure 2.

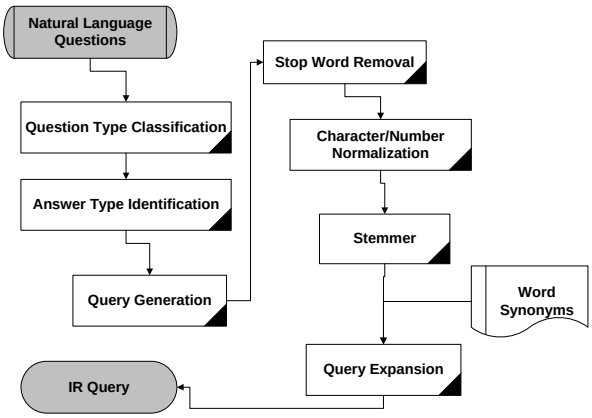


Figure 2. question processing subcomponent

The **document retrieval** component retrieves relevant documents to be used by the sentence/paragraph re-ranking module. For the document retrieval module, different techniques were used from the Lucene contribution package. Lucene is a high performance, scalable IR library. It helps to add indexing and searching capabilities to applications. It is a mature, free, open-source project implemented in Java and a member of the popular Apache Jakarta family of projects, licensed under the liberal Apache Software License. It has facilities for text indexing and searching that can be integrated into applications [17].

SpanNearQuery and RegexQuery of Lucene have been used to maximize retrieval of documents with possible answer particles present. The RegexQuery was specially used to retrieve documents specifically for date and numeric related documents. The SpanNearQuery helps to filter out relevant documents by considering how far the query terms are present in the document. In addition to these techniques, we have also regulated the number of query terms present in the document to be considered relevant to maximize relevant document retrieval. If the number of query terms is less than 3, the document is required to contain all of the query terms to be considered relevant. If the query terms vary from 4 to 6, at least 3/4th of the query terms should be present in the document and if the number of query terms is greater than 7,

the document is required to consist at least half of the query terms to be considered relevant.

The rules that we have designed indicated that as more number of query terms is present in a document, it is a better answer bearing document. Hence, the document retrieval component retrieves the sentence/paragraph and presents these documents to the sentence/paragraph re-ranking module and it also retrieves the total files and present them to the answer selection module.

The **sentence/paragraph re-ranking** module first detects a possible answer particle in the retuned document. We have used two techniques to pinpoint a candidate answer in a document. The first one is Named Entity based (using a gazetteer for place names and person names, and regular expressions for numeric and date question types). The second one is pattern based answer pinpointing where a generic pattern is determined especially for person names. Once answer particles are identified, the best answer is determined based on query term-answer particle distance computation, if multiple answers are detected.

The candidate answer in a document which seems very near to the query terms will be considered possible best answer. Once all possible candidate answers are identified from all documents, then another computation is done based on the number of query terms present in the document. When re-ranking, the document which shows more number of the query terms will be ranked atop, while the one with least number of query terms receives the least rank.

The algorithm we have developed for the re-ranking sentences based on their weight is given in figure 3.

*For each **query term** and **expected answer type (EAT)** accepted from the query generation subcomponent of question processing*
  For each sentence
    *i. Find **answer particle(s)** based on EAT and count the number of answer particles **(count)***
    *ii. If **count > 0** go to **iii**. Else go to **vii**.*
    *iii. For each **answer particle** present*
      *For each **query term***
        *a. Count the number of terms between the **answer particle** and itself*
        *b. Add the value of each query term distance **(distancecount)***
    *iv. If still there are more answer particles, go to **iii**.*
    v. If count=1//only one expected answer
      Return the answer particle
    *vi. Else if count >1, return the answer particle with the least distance **(distancecount)***
    vii. Else if count = 0, discard the sentence //no answer particle presents
    viii. If the selected answer particle
      *Count the number query terms in a sentence **(countqueryterm)***
      *Assign **countqueryterm** as a weight to the answer particle*
  End for
For all sentences with the identified answer particle
  *Sort the sentences based on their weight//based on **countqueryterm***
End for

Figure 3. Algorithm for determining sentence weight

The **answer selection** module selects the best top 5 answers from the previously ranked documents. Besides the already determined rank, the answer selection module also checks for possible repetition of answer particles from the candidate answer pool. If a given answer particle is repeated, the rank of the two will be summed to give a newer rank. Answer particles with the maximum rank value will be selected as an exact answer. The answer selection module considers two answers as equivalent if one is a short form of the other.

For example **ጠቅላይ ሚኒስትር አቶ መለስ ዜናዊ** (Prime Minister Ato Meles Zenawi), **አቶ መለስ ዜናዊ** (Ato Meles Zenawi), **ጠቅላይ ሚኒስትር መለስ** (Prime Minister Meles), and **አቶ መለስ** (Ato Meles) are all considered equivalent. The algorithm for selecting best answer is shown in figure 4.

*While there are extra sentences containing a **candidate answer***
  *For a **candidate answer** in a sentence*
    ***Count** the number of query terms in the sentence //count*
  For a candidate answer in a sentence
    Compare the candidate answer with candidate answers in other sentences
      *If it matches with other candidate answers (or **contains** function)*
        Add count of the two answer particles
        Concatenate the two sentences for summary
        Remove the sentence from the list
  End for
End while
For each weighted sentence //selection
  For each sentence (concatenated sentences) compare its weight with succeeding sentence(s)
    If the weight of the current sentence(s) is less than the succeeding sentence
      Swap their position
    Return the highest count sentence
  End for
End for

Figure 4. Selecting a sentence based on the higher occurrence of a candidate

6. EXPERIMENT

Java Programming language, the Lucene API, and a number of other third-party Java libraries such as *Fileutils* are used in developing our prototype. Figure 5 shows the user interface of our prototype.

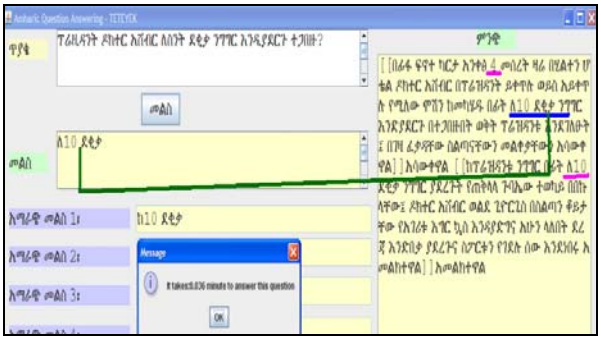


Figure 5. User Interface for AQA

Nearly 12000 question sets have been collected from the Web, Ethiopian Television games and from questionnaire respondents. A total of 15600 Amharic news articles (42 MB) corpus has been collected and normalized. Out of 12000 total questions, nearly 500 factoid questions are selected for the experiment. Hence, the experiment is conducted on the designed sample question and answer sets. The evaluation

criteria we have used were correct answer accuracy. Hence the accuracy of our system is evaluated for recall, precision, percentage, and mean reciprocal rank (MRR). The evaluation methods for these criteria are as follows:

Recall: is calculated as the total number of correct answers over the total of correct and missed answers while present in the corpus.

$$\text{Re}call = \frac{ca}{ca + ma} X100\%$$

where ca is correct answers and ma is missed answers

Precision: is calculated as the percentage of correct answers over the total of correct answers, wrong answers, and No answers.

$$\text{Pr}ecision = \frac{ca}{ca + wa + na} X100\%$$

where ca is correct answers, wa is wrong answers and na is no answers.

Percentage: is calculated as the total number of correct answers over all responses, wrong answers over all responses, and No answers over all responses.

$$Percentage\,of\ Correct\ answers = \frac{ca}{ta} X100\%$$

$$Percentage\,of\ Wrong\ Answers = \frac{wa}{ta} X100\%$$

$$Percentage\,of\ No\ Answers = \frac{na}{ta} X100\%$$

where ta is total answers.

Mean Reciprocal rank (MRR): is also computed to evaluated average rank of answers; where rank is from top one to top five.

$$\text{MRR} = \frac{\sum_{i}^{n} \frac{1}{Ri}}{n}$$

where Ri is the rank of a given answer which ranges from 1 to 5, and n is the total number of answers (correct + wrong + No answer).

The performance of our system has been evaluated before and after document normalization. The evaluation showed the significant effect of document normalization for performance. Consider figures 6 and 7 to see the impact of document normalization.

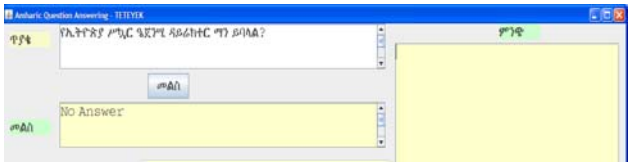


Figure 6. Before charcter normalization

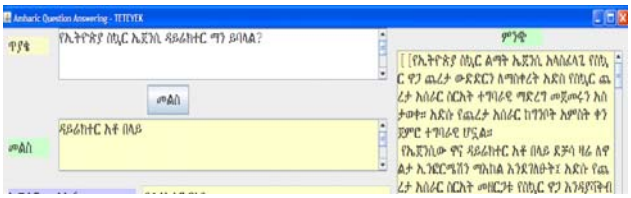


Figure 7. after character normalization

Figure 6 shows that the question **"የኢትዮጵያ ሥኳር ዔጀንሢ ዳይሬክተር ማን ይባላል?"** (Who is the director of Ethiopian Sugar Industry Agency?), has no answer as the characters **ሥ**, **ዔ**, and **ሢ** are not normalized to reflect same character representations as of the document in the corpus. We can clearly see the impact of the character normalization in figure 7 where the same question returns the correct answer.

The effect of document normalization is shown in table 2.

Table 2. Effects of Document normalization

Document	Before normalization		After normalization	
	Precision	Recall	Precision	Recall
Sentence	53.3%	60.3%	66.6%	82.4%
Paragraph	55.4%	63.1%	63.7%	80.6%
File	51.4%	63.9%	55.3%	75.6%

Table 2 shows that document normalization has a performance gain of 7% for precision and 12% for recall.

The Question Answering system correctly classified 89% of the questions using the rule based classification, while 62% are correctly classified by the IR based question classification technique where question sets and answer sets are indexed so that an unseen question will be matched with the help of document similarity computations later.

Table 3. document retrieval performance

Index type	Correct answer particles present	Wrong answer particles present
Sentence	465 (93 %)	35 (7%)
Paragraph	477 (95.4 %)	23 (4.6 %)
File	486 (97.2 %)	14 (2.8 %)

Our document retrieval component also shows an excellent performance as shown in table 3.

Table 4 shows our Named-entity-based answer selection performance for person and numeric question types while table 5 shows pattern based answer selection performance. The pattern based answer selection outperforms the named entity based answer selection technique as the named entity based answer selection technique fails to address all possible answer particles.

Table 4. Gazetteer based correct answer performance

Document	Number of correct answers	Number of wrong answers	Number of No Answers	Missed Answers	Precision	Recall	MRR
Sentence	60 (56.6 %)	30(28.3 %)	16 (15.1%)	11	57%	85%	50%
Paragraph	72 (67.9%)	20 (18.9%)	14 (13.2%)	8	68%	90%	58%
File	60 (56.6%)	34 (32.1 %)	12 (13.3%)	6	57%	91%	44%

Table 5. pattern based answer selection performance

Document	Answer type	Correct answer	Wrong answer	No answer	Missed answers	precision	Recall	MRR
Sentence	Numeric	106(82.8)	18(14.1%)	4(3.1%)	13	83%	90%	71%
	Person	64(60.6%)	22(20.6%)	20(18.8%)	11	60%	85%	50%
	Numeric	78(60.9%)	50(39.1%)	0(0%)	27	61%	74%	52%
paragraph	Person	66(62.3%)	26(24.5%)	14(13.2%)	8	62%	90%	57%
	Numeric	70(54.7%)	56(43.8%)	2(1.7%)	31	55%	69%	45%
file	Person	58(54.7%)	34(32.1%)	14(13.2%)	6	55%	91%	43%

7. CONCLUSION AND FUTURE WORK

This paper attempted to identify the basic language specific issues in question answering for Amharic. The first task we have tackled is normalizing the document so that a standard document is indexed and matching relevant documents during searching are maximized. We have also identified proper question particles as well as question focuses that will help in classifying the question. Gazetteer based and pattern based answer selection algorithms have been developed to maximize correct answer selection. Our algorithm first identifies all possible answer particles in a document. Once the answer particles are identified, the distance of every question particle toward the question terms is calculated. The one with the minimum distance from the query terms is considered the best candidate answer of that document. Once candidate answers are selected from every document, candidate answers which have been repeated more than once (i.e., appeared in more than one document) are given higher ranks.

Candidate answers with maximum number of query terms matched in a document are given higher priority in case a similar rank is given for two or more candidate answers. The evaluation of our system, being the first Amharic QA system, shows very well performance. The rule based question classification module classifies about 89% of the questions correctly. The document retrieval component shows greater coverage of relevant document retrieval (97%) while the sentence based retrieval has the least (93%) which contributes to the better recall of our system. The gazetteer based answer selection using a paragraph answer selection technique answers 72% of the questions correctly which can be considered as promising. The file based answer selection technique exhibits better recall (91%) which indicates that most relevant documents which are thought to have the correct answer are returned. The pattern based answer selection technique has better accuracy for person names using paragraph based answer selection technique while the sentence based answer selection technique has outperformed the performance in numeric and date question types. In general, our algorithms and tools have shown good performance compared with highly resourced language QA systems such as English.

Question answering is a very complex task, which consumes time and needs a number of different NLP tools. Hence, there are a number of rooms for improvement and modification for Amharic question answering. Some of the research works that we have planned to undertake in the future include:

- Developing automatic named entity recognizer
- Incorporating a parser and part of speech tagger
- Developing Amharic WordNet
- Enhancing the Amharic stemmer
- Incorporating Machine learning and statistical Question classifications

REFERENCES

[1] Hu, H. Jiang, P. Ren, F. Kuroiwa, S. 2005. Web-based Question Answering System for Restricted Domain Based of Integrating Method Using Semantic Information Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05. Proceedings of 2005 IEEE International Conference.

[2] Anne-Laure Ligozat, Brigitte Grau, Anne Vilnat, Isabelle Robba, Arnaud Grappy, 2007. Towards an automatic validation of answers in Question Answering, LIMSI-CNRS 91403 Orsay CEDEX France.

[3] Dongfeng Cai Yanju Dong Dexin Lv Guiping Zhang Xuelei Miao, 2004. A web based Chinese Question Answering System, Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05. Proceedings of 2005 IEEE International Conference.

[4] Shouning Qu, Bing Zhang, Xinsheng Yu, Qin Wang, 2008. The Development and Application of Chinese Intelligent Question Answering System Based on J2EE Technology, Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop.

[5] Zheng-Tao Yu Yan-Xia Qiu Jin-Hui Deng Lu Han Cun-Li Mao Xiang-Yan Meng, 2007, Research on Chinese FAQ questions Answering System in Restricted Domain, Machine Learning and Cybernetics, 2007 International Conference.

[6] Jignashu Parikh, M. Narasimha Murty, 2002. Adapting Question Answering Techniques to the Web, Proceedings of the Language Engineering Conference (LEC'02).

[7] Sameer S. Pradhan, Valerie Krugler, Wayne Ward, Dan Jurafsky and James H. Martin, Using Semantic Representations in Question Answering, Center for Spoken Language Research University of Colorado Boulder, CO 80309-0594, USA.

[8] http://en.wikipedia.org/ wiki/Amharic_language, last accessed on October 1, 2008

[9] **ጌታሁን አማረ፣ 1989 የአማርኛ ሰዋሰው በቀላል አቀራረብ** - *Getahun Amare, 1989 Ye-Amarigna sewasew beqelal aqerareb.*

[10] **ባየ ይማም፤1987 የአማርኛ ሰዋሰው፤ ት.መ.ማ.ማ.ድ.** – *Baye Yimam, Ye-Amarigna sewasew, T.M.M.M.D.*

[11] http://www.lonweb.org/link-amharic.htm, last accessed on March 30, 2009.

[12] http://jrgraphix.net/research/unicode _blocks.php?block=31, last accessed on March 31, 2009.

[13] Matthew W. Bilotti, Boris Katz, and Jimmy Lin, 2004, What Works Better for Question Answering: Stemming or Morphological Query Expansion?, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

[14] Cheng-Wei Lee, Cheng-Wei Shih, Min-Yuh Day, Tzong-Han Tsai, Tian-Jian Jiang, Chia-Wei Wu, Cheng-Lung Sung, Yu-Ren Chen, Shih-Hung Wu, Wen-Lian Hsu, 2005, ASQA: Academia Sinica Question Answering System for NTCIR-5 CLQA, Proceedings of NTCIR-5 Workshop Meeting, December 6-9, 2005, Tokyo, Japan.

[15] Tomek Strzalkowski and Sanda Harabagiu, 2008, Advances in Open Domain Question Answering, Published by Springer, ISBN 978-1-4020-4746-6, The Netherlands.

[16] Mark A. Greenwood, 2005, AnswerFinder: Question Answering from your Desktop, Department of Computer Science University of Sheffield Regent Court, Portobello Road Sheffield S1 4DP UK

[17] Otis Gospodnetic, Erik Hatcher, 2005, Lucene in Action, Manning Publications Co., 209 Bruce Park Avenue, Greenwich, CT 06830, ISBN 1-932394-28-1.

[18] Xiaoyan Li and W. Bruce Croft, 2002, Evaluating Question-Answering Techniques in Chinese, In NIST Special Publication: The 10th Text Retrieval Conference.

[19] Praveen Kumar, Shrikant Kashyap, Ankush Mittal, Sumit Gupta, 2005, A Hindi Question Answering system for E-learning documents, Proceedings of the 2005 3rd International Conference on Intelligent Sensing and Information Processing.

[20] Seid Muhie Yimam and Mulugeta Libsie, “TETEYEQ: Amharic Question Answering For Factoid Questions”, In Proceedings of Information Retrieval and Information Extraction for Less Resourced Languages Conference (IE-IR-LRL), Donostia, Spain, September 7th, 2009.