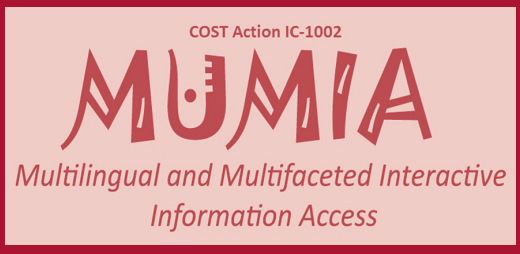
Mihai Lupu
Evangelos Kanoulas
Fernando Loizides (Eds.)

LNCS 8201

Multidisciplinary Information Retrieval

6th Information Retrieval Facility Conference, IRFC 2013
Limassol, Cyprus, October 2013
Proceedings



Lecture Notes in Computer Science 8201

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

Mihai Lupu Evangelos Kanoulas
Fernando Loizides (Eds.)

Multidisciplinary Information Retrieval

6th Information Retrieval Facility Conference, IRFC 2013
Limassol, Cyprus, October 7-9, 2013
Proceedings

Springer

Volume Editors

Mihai Lupu
Vienna University of Technology
Institute of Software Technology and Interactive Systems
Favoritenstrasse 9-11/188, 1040 Vienna, Austria
E-mail: lupu@ifs.tuwien.ac.at

Evangelos Kanoulas
Google Inc.
Brandschenkestrasse 110, 8002 Zurich, Switzerland
E-mail: ekanou@google.com

Fernando Loizides
Cyprus University of Technology
Department of Multimedia and Graphic Arts
30 Archbishop Kyprianou Street, 3036 Limassol, Cyprus
E-mail: fernando.loizides@gmail.com

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-41056-7 e-ISBN 978-3-642-41057-4
DOI 10.1007/978-3-642-41057-4
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013948096

CR Subject Classification (1998): H.3, H.2, I.2.7, H.4, H.5

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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

These proceedings contain the refereed papers presented at the 6th Information Retrieval Facility Conference (IRFC) for Science and Technology, which was held in Limassol, Cyprus, during October 7–9, 2013. The conference was organized by the Cyprus Interaction Lab, Dept. of Multimedia and Graphic Arts of the Cyprus University of Technology and the IC1002 COST Action Multilingual and Multifaceted Interactive Information Access (MUMIA), and endorsed by the Information Retrieval Special Interest Group of the British Computer Society.

IRFC 2013 received 16 quality submission from 12 different countries, of which 8 were accepted as full papers and 2 as short papers. Each submission was reviewed by four Program Committee members. Keeping in line with the objectives of the IRFC to provide a multi-disciplinary scientific forum to bring young researchers into contact with industry at an early stage, this year's program covers a wide range of topics, from fundamental IR issues to domain specific applications. Bhogal and Macfarlane (*Ontology Based Query Expansion with a Probabilistic Retrieval Model*) address a classical problem of IR engines, query expansion, in this case using a news domain ontology. They confirm that consistently better results are difficult to obtain using ontologies for enhancing queries, even in a specialized domain. Shrestha, Vulic, and Moens's work (*An IR-Inspired Approach to Recovering Named Entity Tags in Broadcast News*) may assist in this sense, by identifying specific concepts related to the collection at hand, rather than a general ontology. And also concerning news and news reporting, Chenlo and Losada's article (*A Machine Learning Approach for Subjectivity Classification Based on Positional and Discourse Features*) tells us how to identify human bias in text. Finally, in the category of general IR problems, Audeh and colleagues introduce an interesting modification to an existing IR evaluation measure focusing on the high-recall requirements which exist in specific domains.

Such specific domains are addressed in three of the papers presented this year. First, Hurtado Martin and colleagues present a method to assist researchers in identifying conferences of interest based on their Calls for Papers (*An Exploratory Study on Content-Based Filtering of Calls for Papers*). Then, Moumtzidou and colleagues (*Concept Extraction from Patent Images Based on Recursive Hybrid Classification*) and Andersson and colleagues (*Domain Adaptation of General Natural Language Processing Tools for a Patent Claim Visualization System*) explore the patent domain, together covering both image and natural language processing, to assist patent searchers in identifying the essence of a patent document.

Users are at the center of the other three papers presented this year. Pharo and Nordlie analyze interaction logs in order to estimate the effort invested into the search process (*Using 'Search Transitions' to Study Searchers' Investment of Effort: Experiences with Client and Server Side Logging*). Loizides and Buchanan

propose a framework to closely follow the search behavior of the user. (*Towards a Framework for Human (Manual) Information Retrieval*). Finally, Salampasis and Hanbury discuss what professional search systems should look like and propose a general model, whose individual components may or may not be addressed for different professional search scenarios.

In addition to the papers, IRFC 2013 also hosted a competitive demo organized in collaboration with the TREC Sessions Track. While the systems participating in this demo are not described in these proceedings, the purpose of the competitive demo was to obtain a complementary, qualitative view of human interaction in IR, to complement the predominantly quantitative focus of standardized benchmarks.

We were fortunate to have a global leader keynote speaker, Dr. Ralf Steinberger, whose interests are particularly relevant to the contemporary theme of multilingual and cross-lingual news analysis in the Europe Media Monitor.

Finally, our sincere thanks go out to the Local Organizing Committee; the members of the Program Committee and the additional reviewers for their thorough reviews; the MUMIA COST Action; and the Cyprus University of Technology. The success of IRFC 2013 was also due to our various supporters, but in particular the British Computer Society, the Cyprus Computer Society, and the Cyprus Tourism Organization.

We hope that you enjoy the conference proceedings.

July 2013

Mihai Lupu
Evangelos Kanoulas
Fernando Loizides

Organization

Program Committee

Galia Angelova	Bulgarian, Academy of Sciences, Bulgaria
Avi Arampatzis	Democritus University of Thrace, Greece
Pavel Braslavski	Ural Federal University/Kontur Labs, Russsia
Paul Buitelaar	DERI - National University of Ireland, Galway, Ireland
Pablo Castells	Universidad Autónoma de Madrid, Spain
Ivan Chorbev	Cyril and Methodius University, Macedonia
Bruce Croft	University of Massachusetts Amherst, USA
Ronan Cummins	University of Greenwich, London, UK
Arjen de Vries	CWI, The Netherlands
Sebastien Ferre	Universite de Rennes 1, France
Wilfried Gansterer	University of Vienna, Austria
Gregory Grefenstette	Exalead, France
Allan Hanbury	Vienna University of Technology, Austria
Katja Hofmann	ISLA, University of Amsterdam, The Netherlands
Ivan Koichev	University of Sofia "St. Kliment Ohridski", Bulgaria
Udo Kruschwitz	University of Essex, UK
David Lamas	Tallinn University, Estonia
Christina Lioma	University of Copenhagen, Denmark
Walid Magdy	Qatar Computing Research Institute, Qatar
Edgar Meij	University of Amsterdam, The Netherlands
Igor Mozetic	Jozef Stefan Institute, Slovenia
Henning Müller	HES-SO, Switzerland
Hidetsugu Nanba	Hiroshima City University, Japan
Andreas Nuernberger	University of Magdeburg, Germany
Andreas Rauber	Vienna University of Technology, Austria
Tony Russell-Rose	UXLabs, UK
Michail Salampasis	Vienna University of Technology, Austria
Oren Somekh	Yahoo! Labs, Israel
Dolf Trieschnigg	University of Twente, The Netherlands
Manos Tsagkias	ISLA, University of Amsterdam, The Netherlands
Yannis Tzitzikas	University of Crete and FORTH-ICS, Greece

David Vallet	Universidad Autónoma de Madrid, Spain
Suzan Verberne	Radboud University Nijmegen, The Netherlands
Robert Villa	University of Sheffield, UK
Stefanos Vrochidis	Information Technologies Institute, Greece

Additional Reviewers

Gossen, Tatiana	Pereira, Bianca
Osenova, Petya	Tannebaum, Wolfgang

Table of Contents

Multilingual and Cross-Lingual News Analysis in the Europe Media Monitor (EMM)

(Extended Abstract)

Ralf Steinberger

European Commission – Joint Research Centre (JRC), Ispra (VA), Italy
Ralf.Steinberger@jrc.ec.europa.eu

Abstract. We give an overview of the highly multilingual news analysis system *Europe Media Monitor* (EMM), which gathers an average of 175,000 online news articles per day in tens of languages, categorises the news items and extracts named entities and various other information from them. We explain how users benefit from media monitoring and why it is so important to monitor the news in many different languages. We also describe the challenge of developing text mining tools for tens of languages and in particular that of dealing with highly inflected languages, such as those of the Balto-Slavonic and Finno-Ugric language families.

Keywords: Media monitoring, text mining, multilinguality, information extraction, inflection, highly-inflected languages.

1 Introduction – Overview of EMM, Its Users and Uses

The news analysis system *Europe Media Monitor* (EMM) gathers an average of 175,000 online news articles per day from about 4,000 online sources in up to 75 languages, groups related news articles into clusters, categorises the news items into hundreds of categories, and – for currently 21 languages[1] – extracts named entities and various other information from them. The four EMM applications *NewsBrief*, *NewsExplorer*, the *Medical Information System MedISys* and *EMM-Labs* are publicly accessible via http://emm.newsbrief.eu/overview.html. The pages of *NewsBrief* and *MedISys* (general news vs. Public Health-related news, respectively) always show the very latest news (pages are updated every ten minutes) and display the news in categories. *NewsExplorer*, on the other hand, provides a daily overview and links the news over time and across languages. Additionally, NewsExplorer's entity pages display the historical news related to hundreds of thousands of persons and organisations. EMM-Labs contains applications not yet entirely integrated with the other applications, including event recognition (who did what to whom, where and when), news cluster summaries and a machine translation demo interface.

[1] The 21 *NewsExplorer* languages are: Arabic, Bulgarian, Danish, Dutch, English, Estonian, Farsi, French, German, Hungarian, Italian, Norwegian, Polish, Portuguese, Romanian, Russian, Slovene, Spanish, Swahili, Swedish and Turkish.

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 1–4, 2013.

EMM users include the European Union (EU) institutions, national authorities of the 27 EU Member States, international organisations such as the African Union and various United Nations sub-organisations, individual EU partner countries and the general public (with an average of thirty thousand visitors per day). Most institutional users use EMM to monitor the live media in their specific field of interest, e.g. news on Public Health issues that may be a danger to the public, news on issues that are at the focus of public debate at the moment, or news on crises world-wide (natural disasters, man-made disasters, political conflict, etc.). EMM basically allows users to see media reports of their potential interest on a single page. The display of the extracted meta-information (e.g. search words found, named entities, quotations by and about people, translations, and more) together with the news items allows the users to identify important information more quickly. The news moderation interface *NewsDesk* allows the media departments of organisations to compile readily formatted in-house newsletters to distribute a human digest of the automatically extracted information.

EMM visualises various automatically produced statistics for each news category in graphs to allow the detection and monitoring of trends. An early-warning functionality recognises potentially highly relevant information peaks and automatically informs users of such events by email or by SMS. Since May 2013, EMM is also available as an app for Apple iPad (search iTunes for 'EMM'). A version for Android-based systems is under preparation. For a more detailed overview of EMM, see [5].

EMM thus goes much beyond the functionality of news aggregators such as Google News[2] or Yahoo News[3]. There are other news analysis systems like EMM, but most of these systems are monolingual, including SiloBreaker[4], NewsVine[5] and DayLife[6]. The commercial news analysis system NewsTin[7] with its eleven languages is a notable exception. Monitoring and analysing the news in many different languages is of utmost importance to EMM users as there is ample evidence that news content is highly complementary across languages ([4], [1]). Only the biggest news events are reported around the world, while most smaller events of interest to monitoring the security situation of a country (e.g. on disease outbreaks, criminal activity, conflict situations, etc.) can only be found in national or even regional news.

2 Information Extraction in EMM

EMM not only makes use of Information Extraction (IE) to display meta-information with each news article or news cluster. It also exploits this meta-information to identify news items that are related over time and across languages. For instance, *NewsExplorer* identifies related news articles across its 21 different languages – for all 210 language

[2] See `http://news.google.com`. All websites mentioned here were last visited on 18.06.2013.
[3] See `http://news.yahoo.com/`
[4] See `http://www.silobreaker.com/`
[5] See `http://www.newsvine.com/`
[6] See `http://www.daylife.com/`
[7] See `http://www.newstin.com/` and `http://www.linkedin.com/company/newstin`

pairs – by calculating a cross-lingual news cluster similarity and by using a threshold. News are considered to be equivalent if they (more or less) talk about the same subject domains, if they mention the same persons and organisations (entities), and if the news texts share words across languages (typically names or cognates). For details, see [3].

This cross-lingual similarity calculation relies on the fact that the subject domains and the entity names are the same across languages, or that the equivalences are identified. To determine a ranked list of the major subject domains of the news, we therefore categorise the news automatically according to the EuroVoc thesaurus[8] because the EuroVoc subject domain classification has been translated into all 23 official EU languages (and more) and because manually categorised document collections exist that can be used to train classifiers. We decided to tackle this multi-label classification challenge for a highly unbalanced training set as a profile-based category ranking task ([7]). For named entities (NE), the challenge is to identify that names like Bashar Assad, Bachar al-Assad, Beşşar Esad, Башар Асад and بشار الأسد are all variants of the same name, i.e. that of the current Syrian president. In NewsExplorer, we recognise these name variants automatically through a combination of transliteration, name normalisation and string similarity calculation. The transliteration consists of standard hand-written character n-gram equivalence rules. The normalisation consists of about thirty replacement rules that are the same for all languages and that aim at unifying empirically observed spelling differences, such as for instance the various transliterations of the Russian name suffix –ов (–ov, –ow, –ev, –ew). The string distance similarity – applied only to name variants that have the same normalised form after even deleting the vowels – is performed using either the Levenshtein edit distance or a weighted variant thereof. See ([2]) for further details.

3 Information Extraction and Highly Inflected Languages

When developing Information Extraction tools for the currently 21 NewsExplorer languages, highly inflected languages posed three particular problems: Firstly, recognition patterns (such as identifying uppercase words as person names if they occur next to titles) will not apply if the lexical pattern does not match due to inflection. For instance, the Slovene inflection form *direktorja* found in text will not match the base form *direc-tor*, which is the usual word form found in title dictionaries. The additional word forms thus need to be captured. Secondly, the lookup of uninflected known entities from geographical gazetteers and other name lists will not work when these entities occur in inflected form inside the text. EMM's lists of location, person and organisation names contain over one million entries. Thirdly, when finding new names in text such as Hungarian *Obamáékkal* via Named Entity Recognition (NER), we want to know whether this is the base form or an inflected form of the name. In the latter case, we want to lemmatise the name to its base form before adding the name to our name database (*Obamáékkal* actually is an inflection form of the name *Obama* so that we do not want to add *it as a separate entry* to the database). In EMM, we have no access to morphologi-

[8] See `http://eurovoc.europa.eu/`

cal analysers or lemmatisers because we try to stay independent of third-party software and we try to keep the processing pipeline the same for all languages. We therefore work with Kleene star type wildcards wherever possible and we use hand-crafted rules to pre-generate inflection forms of dictionary entries or to lemmatise names using a similar kind of rules. This helps us capture more entries and to solve some of the problems, but we need to intensify our efforts as we feel that more could be done to cover highly inflected languages better. [6] discusses these inflection-related issues in more detail, as well as the impact of highly inflected languages for document categorisation.

References

1. Piskorski, J., Belyaeva, J., Atkinson, M.: Exploring the usefulness of cross-lingual information fusion for refining real-time news event extraction. In: Proceedings of the 8th International Conference on Recent Advances in Natural Language Processing (RANLP 2011), Hissar, Bulgaria, September 12-14, pp. 210–217 (2011)
2. Pouliquen, B., Steinberger, R.: Automatic Construction of Multilingual Name Dictionaries. In: Goutte, C., Cancedda, N., Dymetman, M., Foster, G. (eds.) Learning Machine Translation, pp. 59–78. MIT Press - Advances in Neural Information Processing Systems Series, NIPS (2009)
3. Pouliquen, B., Steinberger, R., Deguernel, O.: Story tracking: linking similar news over time and across languages. In: Proceedings of the 2nd Workshop on Multi-source Multilingual Information Extraction and Summarization (MMIES 2008) Held at CoLing 2008, Manchester, UK (2008)
4. Steinberger, R.: A survey of methods to ease the development of highly multilingual Text Mining applications. Language Resources and Evaluation Journal 46(2), 155–176 (2012)
5. Steinberger, R., Pouliquen, B., van der Goot, E.: An Introduction to the Europe Media Monitor Family of Applications. In: Gey, F., Kando, N., Karlgren, J. (eds.) Information Access in a Multilingual World - Proceedings of the SIGIR 2009 Workshop (SIGIR-CLIR 2009), Boston, USA, pp. 1–8 (2009)
6. Steinberger, R., Ehrmann, M., Pajzs, J., Ebrahim, M., Steinberger, J., Turchi, M.: Multilingual media monitoring and text analysis – Challenges for highly inflected languages. In: Habernal, I., Matoušek, V. (eds.) TSD 2013. LNCS (LNAI), vol. 8082, pp. 22–33. Springer, Heidelberg (2013)
7. Steinberger, R., Ebrahim, M., Turchi, M.: JRC EuroVoc Indexer JEX - A freely available multi-label categorisation tool. In: Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, pp. 798–805 (2012)

Ontology Based Query Expansion with a Probabilistic Retrieval Model

Jagdev Bhogal[1] and Andrew Macfarlane[2]

[1] Birmingham City University,
Faculty of TEE,
School of CTN,
Millennium Point,
Curzon Street,
Birmingham B4 7XG, UK
[2] City University London,
Centre for Interactive Systems Research,
School of Informatics,
Northampton Square,
London EC1V 0HB, UK

Abstract. This paper examines the use of ontologies for defining query context. The information retrieval system used is based on the probabilistic retrieval model. We extend the use of relevance feedback (RFB) and pseudo-relevance feedback (PF) query expansion techniques using information from a news domain ontology. The aim is to assess the impact of the ontology on the query expansion results with respect to recall and precision. We also tested the results for varying the relevance feedback parameters (number of terms or number of documents). The factors which influence the success of ontology based query expansion are outlined. Our findings show that ontology based query expansion has had mixed success. The use of the ontology has vastly increased the number of relevant documents retrieved, however, we conclude that for both types of query expansion, the PF results are better than the RFB results.

Keywords: Ontology, Query Expansion, Probabilistic Retrieval Model, Okapi, relevance feedback, pseudo-relevance feedback.

1 Introduction

In traditional information retrieval (IR) systems, the search process was iterative. Relevance feedback information was taken from the user so the retrieval process could be repeated using the additional relevance information. However since the users might be reluctant to provide feedback, researchers started focusing on contextual IR [1]. Contextual IR integrates the user context into the retrieval process. Context can be inferred in many different ways. An ontological model can effectively disambiguate meanings of words from free text sentences [3]. An ontology is a collective body of knowledge

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 5–16, 2013.

which is usually created and shared by users who are experts in that domain. Ontologies can be used to infer context for ambiguous queries. The concepts in the ontology can be used for word sense disambiguation and subsequent query expansion.

A collection independent ontology is used for our experiments and ontology based query expansion is applied to the news domain. The ontological approach is suitable for the information intensive news domain. News is the communication of information on current events which is presented by print, broadcast, internet or word of mouth to a third party or mass audience. A news ontology is usually created and shared by a group of specialists in the news field such as journalists, editors and Press standards organisations. Domain specific ontologies are used to model specialised vocabulary from that field such as medical terms. The news domain doesn't have a specific vocabulary as such it just uses plain English language in an accepted journalistic style. However what is important within this domain is the structure of news items. The structure of a news item includes: Headline, subheading, date, author, event description. News ontologies can be used to assist in different tasks such as news categorisation/classification, reasoning; searching; news annotation; updating, news summarization and news alerts. The chosen ontology has been derived from news articles so it is appropriate to use it for the searching task on the TREC document collection.

The paper is structure as follows. We state the motivation for the research and our aims. The methodology including process and collection used to carry out the research is then described. The system used to undertake experiments is outlined. Experimental results are reported with a discussion on the implications of the results, comparing with similar studies. We provide an overall conclusion and pointers to further research in the area.

2 Motivation and Aims

This paper seeks to address questions such as whether the use of query expansion increases recall, precision or both and secondly how ontology based query expansion compares with relevance feedback/pseudo-relevance feedback techniques. This is the first time this particular TREC document collection and news ontology are being used in conjunction with each other so the results produced can provide useful baseline statistics for other researchers who want to carry out retrieval experiments using this particular combination of document collection and ontology.

The paper attempts to combine both approaches of relevance feedback query expansion and ontology based query expansion. A detailed investigation is carried out into the area of query expansion using a news ontology in a probabilistic retrieval environment. Since we are interested in the news domain, an appropriate document collection and domain-specific ontology is selected. We use pseudo relevance and relevance feedback techniques [12] and utilise relevance feedback information from pre-stored relevance judgments which indicate for each document whether it is relevant to the topic query or not. The techniques have proved to be successful to a certain extent - the revised retrieval model will build on the existing Robertson/

Spark-Jones retrieval model [12] and incorporate the use of the ontology information into the query expansion process.

Our research experiments test whether the ontology has a positive impact on relevance feedback and also what effect varying the relevance feedback parameters on the query expansion results.

3 Methodology

We expand all queries and do not attempt to assess their ambiguity. With regards to term selection, all query terms are used for query expansion. Each query term in the index is searched to provide new query terms, however in addition to this, the parent-child database is searched to provide ontology based query expansion terms.

Short queries are better candidates for query expansion because they have insufficient terms to describe the information need and tend to be more ambiguous [9]. Therefore the query files are based on the topic titles (defined in section 3.2) only because they form shorter queries compared to queries based on the topic description.

The two main parameters of relevance feedback are: selection of terms and the sample size of relevant documents. In the Okapi system traditionally these have been 20 terms and 20 documents. Billerbeck and Zobel [2] state that the choice of query expansion parameters used can affect the retrieval performance. As part of this research we experiment in varying these relevance feedback parameters and analyse the impact on the results. Another question that is addressed is whether to use all expanded terms or select the top 3 query expansion terms.

3.1 Description of the Ontology

The WNO ontology was created by Kallipolitis et al [7] who studied a large number of international news articles from news agency websites and as a result based the ontology on 11 subjects which they felt were sufficiently representative in the domain of world news. The WNO ontology adheres to a tree structure with a maximum depth of 2 levels with class information provided in a top down fashion. The size of the WNO is 29.Kb making it easier to navigate and process programmatically.

WNO is written in XML in News Industry Text Format (NITF) which is published by IPTC and is designed to standardize the content and structure of text-based news articles; xml enhances system portability; WNO is relatively easy to process and is based on the industry standards news codes taxonomy produced by ITPC - NewsML, [10]. NewsML provides a set terms for the news domain. This set of terms also known as Newscodes includes a hierarchy of terms and concepts that can be used to describe news in any field of interest.

Attributes are stored for each of the levels in order to enrich certain terms and make their meaning more specialized. The attributes are useful in the search process for strengthening the similarity match between different topics. Our research does not currently make use of the attribute information but it is recommended for future use.

3.2 Test Collection

We selected the TREC newswire document collection from TREC (Disk2) because it is a reasonable size (over 231,000 documents) and even though it is not as large as other document collections it has associated topics/queries and also the relevance judgements were readily available [6]. Therefore it is ideal to use as a test collection for information retrieval evaluation. TREC document collections are widely accepted by the information retrieval research community. We used the adhoc task and topics 51-300 used (250 topics in total). A TREC topic is a natural language statement of information need written by real users of retrieval systems. Topics are distinct from queries because they contain more detail than the latter. Queries are constructed from topics and submitted to the retrieval system.

Disk2 is a smaller collection size in comparison to other document collections but the advantage of Disk 2 is that it contains a wider range of topics. The document collection contains news articles and non-news based articles. News based articles were not separated out and the entire collection was used because the aim was to use the as many relevance assessments in the document collection as possible. The non-news articles in the collection introduced "noise" to discover whether the news ontology ranked news articles higher than non-news. There is no strong evidence to suggest that the news ontology favours news over non-news articles possibly because we are not putting any emphasis on the structure of the articles. Only key terms are being used for the search thus we are treating all articles whether they be news or non-news in the same manner. If any structural feature of news articles are incorporated in the search process then it is likely that news articles would appear higher up in the ranked set of results.

3.3 Experiment Design

A set of experiments were designed to answer the research question "Do ontologies have a positive effect on the impact of relevance feedback"? We also wanted to test the effect on the results of varying the relevance feedback parameters. The document collection is indexed in Okapi which uses the probabilistic retrieval model [12]. The document collection is indexed on the TREC document id (DOCNO), heading (HEAD) and description (TEXT) fields. Additionally, the News ontology is searched and hierarchical node relationship information is recorded in a parent-child database. The new system employs RFB and PF techniques but in expands the query further by making use of the parent-child information obtained from the ontology. The parent node(s) of a query term will broaden the query and the child node(s) of a query term will make a query more specific. The two main parameters of relevance feedback are: selection of terms and the sample size of relevant documents. We investigate the effect of varying the number of terms/documents relevance feedback parameters (Table 1).

In the Okapi system traditionally these have been 20 terms and 20 documents [12]. Billerbeck and Zobel [2] state that the choice of query expansion parameters used can

Table 1. Summary of Experiments

Purpose of experiment	Experiment Number
Test ontology based query expansion compared to original system	Experiment 1 uses standard relevance feedback parameter values of 20 documents and 20 terms
Test the effect of varying the number of terms relevance feedback parameter	Experiments 2, 3, 4, 5, 6 use term relevance feedback parameters of 5, 10, 15, 100 and 200 respectively
Test the effect of varying the number of documents relevance feedback parameter	Experiments 7, 8, 9, 10, 11 use document relevance feedback parameters of 5, 10, 15, 100 and 200 respectively
Test the effect of selecting a subset of the expanded terms	Experiment 12 uses standard relevance feedback parameter values of 20 documents and 20 terms but only selects the top 3 expansion terms

affect the retrieval performance. There is no real consensus on the optimum number of documents to use for Query expansion. Sparck-Jones [16] used 20, Robertson et al [14] used 1000 (too much effort for very little return). Search routines were developed which used relevance feedback for query expansion and the resulting set of expanded terms were expanded even further by using associated broader and narrower ontological terms. Experimental results were evaluated using retrieval effectiveness metrics. The resulting collection set could be used by researchers for future experimentation.

3.4 Metrics

Different types of evaluation metrics are required to evaluate the performance of each retrieval model and conduct a comparison. Recall and Precision are single-value metrics which evaluate the quality of an unordered set of documents returned by the system. For systems that return a ranked sequence of documents, it is desirable to also consider the order in which the returned documents are presented. Three other metrics used are mean average precision (MAP), Bpref and precision-recall curves. T-tests are carried out on Document level averages, Precision-Recall , MAP, and Average Recall statistical data to measure the statistical significance of these results. These measures are commonly used by other information retrieval systems thus making it easier to compare our results against those of other systems.

4 System

Okapi is an experimental IR system, written to examine various aspects of interactive IR research, including such tasks as bibliographic search and full-text search [12].

The system uses the Probabilistic Retrieval Model and BM25 weighting functions are used to rank the documents [12] BM25 is a best match operator which retrieves relevant documents higher up the rank based on normalized document length and is the baseline to which other ranking functions are now compared. The probabilistic retrieval model is a highly effective retrieval model that makes explicit distinctions between occurrences of terms in relevant and non-relevant documents [12]. It calculates the probability of a document being relevant if it contains certain terms. A single processor Sun SS10 with 64MB of core and about 12GB of disk was used as the main development machine and file server.

We build a separate database containing semantic information such as parent-child relationships between ontology nodes. This was required so we could transfer the ontology knowledge in an appropriate format and make it accessible to the Okapi software. This information is used to supply additional terms for expanding the original query terms. The semantic parent-child information is stored in memory using a list structure.

5 Experiment Results

The ontology has a better effect higher up the rank for the PF runs of Document Level Averages metric and the RFB runs starts improving from the lower end of the ranked set of documents which implies that the PF runs have more to gain from varying the relevance feedback parameters and do benefit from the use of the ontology. With the RFB runs, use of the ontology based terms for query expansion distorts the retrieval of relevant documents and is only useful at the lower end of the ranked list. In our view the RFB is harder to improve on because the top N documents used for RFB are already judged to be relevant so RFB without the use of the ontology produces good results which are hard to improve on. For the PF runs, the top N documents are **assumed** to be relevant because they are ranked highly by the system. These documents might not contain as many relevant query expansion terms as the RFB documents so any relevant additional ontology based query expansion terms will result in an improvement.

The use of ontology based query expansion has achieved high Recall results. This is possibly because query topics have a higher number of hits in the ontology for broader searches and for each hit, few ontology terms are retrieved but a higher proportion of the terms retrieved are relevant compared to ontology terms retrieved for narrower searches. An explanation for this is that quite a large number of query topic terms are being found in the ontology and even though each of these only has one parent node associated, the use of these parent nodes is retrieving more relevant documents. Sometimes when searching for parent nodes, the ontology produces relevant terms. In other cases the ontology produces non-relevant terms which have a negative effect on precision and recall as shown in the example below:

TOPIC NUMBER = 90 (“data proven reserves oil natural gas producers”)
current word is oil
--> economy_business_finance

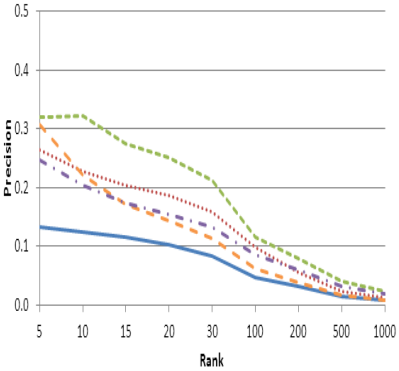
In this case the query topic is more to do with oil as an energy source and has nothing to do with economy_business_finance.

The use of ontology based query expansion has only increased mean average precision for a few cases but overall the precision is usually identical to the baseline or sometimes even below the baseline. In "narrower searches", fewer query topic terms are matched with the ontology terms. Where a match occurs, the ontology term tends to have many more child terms associated with it but the precision-recall depends on the number of child terms that are relevant to the query topic and the number of relevant documents that contain the child term. In a few cases a larger number of relevant results are produced by the ontology which results in improved precision-recall. However in most other cases, just because an ontology term has lots of associated child terms, does not necessarily mean that the number of relevant documents retrieved will increase vastly. For example in narrower searches, where a term is quite general, many child nodes are retrieved of which only one or two might be relevant. Alternatively, the term produced is so general it does not improve the precision results at all because it retrieves a large number of documents which contain the general term and many of these documents are not relevant to the query topic. Another example to illustrate lack of improvement in performance retrieval is where many of the child terms are relevant to the search term but not relevant to the query. Therefore the use of ontology based query expansion has only increased mean average precision for a few cases but overall the precision is usually identical to the baseline or sometimes even below the baseline. The reason for this is that more ontology child terms are retrieved but a smaller proportion of these are actually relevant, thus having minimum impact on precision.

Retrieval results have improved with the use of the ontology but there is no clear trend that increasing the number of terms/documents results in improved retrieval. The number of terms parameter for relevance feedback benefits the PF and PF with ontology results but the number of documents parameter also has an effect on the RFB results. For example the graphs for document level averages (Figure 1) show more of an improvement compared to the Precision-Recall graphs (Figure 2). The reason for this is that it is easier to achieve improvements in precision in the top 5 or top 10 documents compared to achieving improvements in precision at recall .10 especially if the document collection is large. For example if the document collection is 20,000 documents, 0.10 recall calculates to 2000 documents.

Improved retrieval results depend on the ontology coverage of the topic in breadth and depth; the similarity of terms between the ontology and the document collection; and finally the document collection coverage of the ontology terms. The ontology could have a lot of terms relevant to the topic but these terms might not be contained in many documents thus there is minimum impact on performance.

The ontology results in improvements for some topic-sets but not for others. First of all, when searching an ontology using query topic terms, we need to find at least one hit in the ontology for any improvements to take place. Some topics have more ontology hits than others. The second success factor relies not on just the number of hits in the ontology but on the retrieved ontology terms being relevant to the query topic.



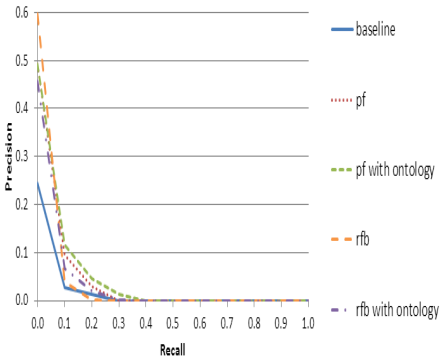


Fig. 1. Topic 51-100 Results Document Level **Fig. 2.** Topic 51-100 Results Precision-Recall Averages

We use stemmed keywords when searching the ontology, so its possible that the actual ontology hits are irrelevant and/or the retrieved ontology terms are irrelevant.

Topic no = 223 (responsible great emergence Microsoft computer industry)
ORIG WORD IS emergenc
--> explosion

Emergence has been stemmed to emergenc, and ontology picks up non-relevant term explosion which is more related to emergency than emergence.

Suppose we find a good set of ontology terms to expand the query with, then the next factor in improving retrieval relates to finding enough documents in the document collection that contain the ontology term and are relevant to the query topic. If the match between the ontology and the document collection is poor, then even though the ontology terms are relevant to the query topic, because there aren't enough documents containing that term, query expansion has minimal effect on recall/precision. Alternatively, if the parent/child term obtained from the ontology is too general, then many documents are retrieved but very few of these are relevant to the query topic.

A topic hardness measure is calculated as the average over a given set of runs of precision for each topic after all relevant documents have been retrieved OR after 100 documents have been retrieved if more than 100 documents are relevant. The measure is oriented towards high-recall performance and how well systems do at finding all relevant documents. If no system does well on a query then it can be called a hard query. According to TREC hardness measure given in Buckley et al (1996) the performance for TREC 4 (topic 201-250) and TREC 5 (251-300) drops from 0.676 to 0.672 and 0.556 respectively. These are seen to be difficult topics because they are progressively shorter in length and higher level in nature. This trend is mirrored in the SMART experiments. For example in TREC1 the precision is 0.2431 and in TREC2 the best precision has improved to 0.2594 but in TREC 4 and TREC 5 the precision dropped to 0.1507 and 0.1038 respectively.

Table 2. Overall Results Summary and Statistical Significance

Experiment	Doc level Averages		Precision-Recall		Recall		MAP		BPref	
	PF	RFB	PF	RFB	PF	RFB	PF	RFB	PF	RFB
5 terms	51-100 **(0.0001)**	251-300	251-300	51-100	51-100 **(0.001)**	51-100 **(0.002)**	51-100	51-100	51-100 (0.005)	51-100
10 terms	251-300 (0.013)	251-300	251-300	51-100	151-200 **(0.004)**	51-100 (0.002)	251-300	51-100	251-300 **(0.000)**	51-100
15 terms	51-100 **(0.003)**	51-100	251-300	51-100	151-200 **(0.006)**	201-250 **(0.003)**	251-300	51-100	251-300 **(0.001)**	51-100
100 terms	251-300 **(0.002)**	51-100 (0.034)	251-300	251-300	251-300 (0.044)	201-250 (0.010)	251-300	51-100 (0.022)	251-300 (0.045)	51-100
200 terms	51-100 **(0.000)**	151-200	251-300	151-200	151-200 **(0.001)**	151-200 **(0.004)**	51-100 **(0.008)**	151-200	101-150 **(0.004)**	101-150
5 docs	251-300 (0.021)	151-200	251-300	51-100	151-200	151-200	251-300	151-200	251-300 **(0.003)**	51-100 **(0.008)**
10 docs	101-150	251-300	251-300	51-100	51-100	51-100	251-300	51-100	51-100	51-100
15 docs	251-300	251-300	251-300	51-100	151-200	51-100	251-300	51-100	251-300	101-150
100 docs	51-100 **(0.002)**	201-250 **(0.008);** 251-300 **(0.009)**	251-300	51-100	51-100	151-200	51-100	251-300 (0.11)	51-100 (0.029)	51-100
200 docs	251-300 **(0.000)**	151-200 **(0.007)**	251-300	251-300	251-300	251-300	101-150	251-300 **(0.008)**	101-150 **(0.003)**	51-100 (0.011)
20 terms/docs	51-100 (0.023)	51-100	251-300	51-100	201-250 (0.013)	201-250 (0.049)	251-300 (0.013)	51-100 **(0.008)**	251-300 **(0.001)**	51-100 **(0.009)**
Top 3 expansion terms	51-100 **(0.001)**	151-200	251-300	51-100	51-100 **(0.000)**	51-100 **(0.001)**	51-100 **(0.001)**	51-100 **(0.000)**	51-100 **(0.000)**	51-100 **(0.001)**

For each run shown in table 2, we compared across the various metrics to see which topics occur the most. Again, topics51-100 and topics251-300 have the highest frequency across the various metrics. According to Buckley et al [4], topic251-300 is considered to be a hard topic set. So the ontology seems to have improved the retrieval performance for a hard topic as well those considered not to be hard.

We can analyse table 2 for statistical significance. For the Document level averages, twice as many PF results are significant/very significant compared to the RFB results. For the Precision-Recall curves, only the PF results are significant/very significant. Recall is the metric with the highest number of statistically significant results.

So we have high recall at expense of precision. This is good for some domains because professional searchers such as investigative journalists prefer to obtain as much information about a given news story as possible. Lawyers need to look at all case statutes in order to produce a strong argument otherwise missed case articles will weaken their evidence. In the same way investigative journalists need to ensure they have accessed all relevant articles in order to produce a thorough report on the subject they are investigating otherwise they will be open to criticism if gaps in the research are found. Also the results analysis shows the document level average results are better than recall –precision and the document level averages (PF runs) are benefitting from the ontology higher up the rank. Again this would indicate that the ranking

algorithm is working and searchers tend to concentrate on the documents occurring higher up in a ranked set of results. The documents for PF are "assumed" to be relevant because they appear high up in the system ranking, whereas the documents for RFB are judged by human assessors as actually being relevant. It would be difficult to improve retrieval performance on the RFB relevant documents; however the PF runs have more to gain from these other factors than RFB.

6 Discussion

It is important to compare our findings with those of related research. Robin and Ramalho [15] used disk2 of the TREC collection and the WordNet ontology to expand query words with some of their synonyms and hypernyms. For comparison purposes, the document collection is the same but we have used a news based ontology to obtain synonyms and hypernyms instead. The other difference is that Robin and Ramalho [15] used the F-measure metric instead of BPref. Finally they used bounds of 10, 15, 20, 30 and 50 documents, we used 5, 10, 15, 20, 100 and 200 terms/documents. They found that all expansion strategies improve overall effectiveness by improving recall more than they worsen precision (in relative terms). Their results show that recall can be boosted by as much as 72.4% relative to the no expansion case. They also expand to the first-level in the ontology. Their best query expansion strategy yields only a 2.51% improvement reaching 9.3% and only 11% of all relevant documents together with 77.5% irrelevant ones. For bounded precision for the top 20, 30, 40 and 50 documents, precision respectively improved by 1%, 12%, 17% and 37%. In comparison to Robin and Ramalho's work [15], our results show that even though use of synonyms or hypernyms is not made there is an average increase of 88% in recall for pseudo relevance feedback and 20% increase in recall for relevance feedback. Unlike Robin and Ramalho [15] however, we did not discover any linear trend resulting from increasing the number of terms/documents.

The use of ontologies for query expansion has had mixed success [5] because they are effective in increasing recall and less successful than RFB but as good as PF [2]. Our findings support these statements. Our attempts at ontology based query expansion have had mixed success. Use of the ontology has vastly increased the number of relevant documents retrieved. We can conclude that for both types of query expansion, the PF results are better than the RFB results. Our findings are similar to that of Billerbeck and Zobel [2] in that ontology based query expansion enhances recall, and produces bigger improvements for PF compared to RFB. The ontology has a better effect higher up the rank for the PF runs of Document Level Averages metric and the RFB runs starts improving from the lower end of the ranked set of documents which implies that the PF runs have more to gain from varying the relevance feedback parameters and do benefit from the use of the ontology. Query expansion seems to be more successful only on relevant documents [2],[11]. In support of this statement, use of the ontology based terms for query expansion in RFB runs is distorting the retrieval of relevant documents and is only useful at the lower end of the ranked list.

7 Conclusion and Further Research

Query expansion has been successful to a certain extent but there is still scope to improve the techniques for selecting and designing algorithms for optimum parameter choice and only expanding queries which would benefit from the query expansion process. The work presented in this research provides some indication of how this can be done and for what type of query expansion.

Our work can be improved by conducting further research on better term selection. Selecting query expansion terms based on relatedness to the whole query is more effective [8]. In TREC 8 [13], a term selection measure was used for selective expansion to measure the statistical significance of any given term's association with relevance.

To increase intelligence, the system should recognise synonyms and utilise homography - a spelling method that represents every sound by a character. Our system does not at present have these features. Compound words add complexity to the query expansion process however; further research is needed on the effective deployment of compound words in query expansion. The WNO ontology does record attributes which could be used in future experiments because attributes give advanced modelling and evaluation options. In addition the ontology should have automated reasoning for creating new concepts based on those that already exist and make further use of inferencing rules in the search process.

Finally we could apply our query expansion algorithms to different ontologies to see what difference each ontology makes to the query expansion process and the reasons why one ontology is inherently better than another. For example the NEWS ontology [17] is larger in size which indicates it has more coverage of the news domain. It also has a more complex lattice structure and deeper levels of nodes than the ontology we used. It would be more complex to process but could produce enriched results.

References

1. Bhogal, J., MacFarlane, A., Smith, P.: A review of Ontology based query expansion. Information Processing and Management 43(4), 866–886 (2007)
2. Billerbeck, B., Zobel, J.: Questioning Query Expansion: An Examination of Behaviour and Parameters. In: Schewe, K., Williams, H.E. (eds.) Proceedings of ADC 2004, pp. 69–76 (2004)
3. Buckland, M.: Translingual information management using domain ontologies (2003), `http://metadata.sims.berkeley.edu/GrantSupported/tides.html` (accessed May 20, 2013)
4. Buckley, C., Singhal, A., Mitra, M.: Using Query zoning and correltation within SMART: TREC-5. In: Fifth Text Retrieval Conference, pp. 105–118. NIST Special Publication 500-238 (1997)
5. Gonzalo, J., Verdejo, F., Chugur, I., Cigarran, J.: Indexing with WordNet synsets can improve text retrieval. In: Boitet, C., Whitelock, P. (eds.) Proceedings of Coling-ACL 1998, Canada, pp. 38–44 (1998)

6. Harman, D.: Overview of the First Text REtrieval Conference. In: Korfage, R., Rasmussen, E., Willett, P. (eds.) Proceedings of SIGIR 1993, pp. 36–48. ACM (1993)
7. Kallipolitis, L., Karpis, V., Karali, I.: World News Finder: How we Cope without the Semantic Web. In: Devedžic, V. (ed.) Proceedings of AIA 2007, pp. 549–221 (2007)
8. Mandala, R., Tokunaga, T., Tanaka, H.: Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion. In: Hearst, M., Gey, F., Tong, R. (eds.) Proceedings of SIGIR 1999, pp. 191–197 (1999)
9. Navigli, R., Velardi, P.: An analysis of Ontology-based query expansion strategies. In: Workshop on Adaptive Text Extraction and Mining - ATEM 2003 (2003)
10. NewsML. Metadata Taxonomies for news (2008), `http://www.iptc.org/site/NewsCodes/` (accessed May 20, 2013)
11. Ogawa, Y., Mano, H.: Selecting expansion terms in automatic query expansion. In: Croft, W., Harper, D.J., Kraft, D.H., Zobel, J. (eds.) Proceedings of SIGIR 2001, pp. 390–391 (2001)
12. Robertson, S.E.: Overview of the Okapi projects. Journal of Documentation 53(1), 3–7 (1997)
13. Robertson, S.E., Walker, S.: Okapi/Keenbow at TREC-8. In: Eighth Text Retrieval Conference, pp. 151–162. NIST Special Publication 500-246 (1999)
14. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gatford, M., Payne, A.: Okapi at TREC-4. In: Fourth Text Retrieval Conference, pp. 73–96. NIST Special Publication 500-236 (1995)
15. Robin, J., Ramalho, F.: Can ontologies improve web search engine effectiveness before the advent of the semantic web? In: Laender, A.H. (ed.) Proceedings of SBBD 2003, pp. 157–169 (2003)
16. Sparck-Jones, K.: Experiments in Relevance Weighting of Search Terms. Information Processing and Management 15(3), 133–144 (1979)
17. Sanchez-Fernandez, L., Fernandez-Garcia, N., Bernardi, A., Zapf, L., Penas, A., Fuentes, M.: An experience with Semantic Web technologies in the news domain. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) Proceedings of ISWC-2005 - Poster (2005)

A Machine Learning Approach for Subjectivity Classification Based on Positional and Discourse Features

Jose M. Chenlo and David E. Losada

Centro de Investigación en Tecnoloxías da Información (CITIUS)
Universidad de Santiago de Compostela, Spain
{josemanuel.gonzalez,david.losada}@usc.es
http://www.gsi.dec.usc.es/ir

Abstract. In recent years, several machine learning methods have been proposed to detect subjective (opinionated) expressions within on-line documents. This task is important in many Opinion Mining and Sentiment Analysis applications. However, the opinion extraction process is often done with rough content-based features. In this paper, we study the role of structural features to guide sentence-level subjectivity classification. More specifically, we combine classical n-grams features with novel features defined from positional information and from the discourse structure of the sentences. Our experiments show that these new features are beneficial in the classification of subjective sentences.

Keywords: Information Retrieval, Opinion Mining, Subjectivity Classification, Sentiment Analysis, Machine Learning, Rhetorical Structure Theory.

1 Introduction

With the advent of the social web, opinions have become a key component in many on-line repositories [1]. These new opinion-oriented resources demand advanced classification processes able to skim off the opinionated texts to reveal the subjective parts. Extracting opinions from text is challenging and poses many problems that cannot be merely solved with lexicon-based approaches. These difficulties are caused by the subjectivity of a document being not so much conveyed by the sentiment-carrying words that people use, but rather by the way in which these words are used. We argue that the study of sentence positional information and intra-sentence discourse structure can help to tackle this issue. For instance, people tend to summarise their viewpoints at the end of the text. Moreover, the rhetorical roles of text segments can effectively guide the opinion detection process. For example, the sentence *"Nevertheless it is undeniable that economic disparity is an important factor in this ethnic conflict"* contains an attribution relationship between the nucleus of the sentence (*"that economic disparity is an important factor in this ethnic conflict"*) and its satellite (*"Nevertheless it is undeniable"*). The presence of this relation helps to understand

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 17–28, 2013.

that the writer is expressing his/her point of view (satellite) about the statement presented in the nucleus. This type of rhetorical clue is potentially valuable to detect opinions.

In this paper we combine bag of words features, such as unigrams or bigrams, with features computed from sentiment lexicons and with more advanced positional and rhetorical features. Our results show that the combined use of these features leads to an accurate classification of subjective sentences. To the best of our knowledge, this is the first attempt to combine rhetorical, content-based and positional features for a fine-grained (i.e., sentence-level) estimation of subjectivity. As argued in our related work section, other studies have explored the role of rhetorical features in Opinion Mining (OM) but previous efforts are mostly based on coarse-grained tasks (e.g., categorising the overall orientation of a movie review). Document-level sentiment classification is too crude for most applications and we need to move to the sentence level to support a more advanced analysis of sentiments [2].

2 Sentence Features

We focus on a two-class (subjective vs. non-subjective) classification of sentences[1] and take into account the following traditional and advanced features to build our classifiers:

Unigram & Bigram Features. These are binary features based on the appearance of unigrams and bigrams in the sentence[2].

Sentiment Lexicon Features. These features are based on counting the sentiment-bearing terms that occur in the sentence. The sentiment lexicon was obtained from OpinionFinder [3], which is a well-known subjectivity classifier. We include the number and percentage of opinionated terms in a sentence as features for our classifiers. We also consider the number and percentage of interrogations and exclamations in the sentence. The ability of these features to detect opinions has been demonstrated in other studies [4].

Rhetorical Features. Rhetorical Structure Theory (RST) [5] is one of the leading discourse theories. This theory explains how texts can be split into segments that are rhetorically related to one another. Within this structure, text segments can be either nuclei or satellites, with nuclei being assumed to be more significant than satellites with respect to understanding and interpreting a text. Many types of relations between text segments exist; the main paper on RST defines 23 types of relations [5]. A satellite may for instance be an elaboration, an explanation or an evaluation on what is explained in a nucleus. We used

[1] A subjective sentence may not express any sentiment ("I think that he left") and an objective sentence can imply an opinion ("The phone broke in three days") [2]. We are interested in detecting opinions at sentence-level and, therefore, we search for sentences expressing either explicit or implicit sentiments. However, we use subjective and opinionated interchangeably to refer to sentences that implicitly or explicitly express opinions.

[2] Unigrams and bigrams with less than 4 occurrences in the collection are removed.

SPADE (Sentence-level PArsing of DiscoursE) [6], which creates RST trees for individual sentences and we include binary features associated to the appearance of every type of RST relation in a given sentence. Observe that we make an intra-sentence RST analysis. The study of inter-sentence RST analysis is an interesting challenge that is out of the scope of this paper. The relation types considered are reported in Table 1.

Length Features. These features encode the length of the sentence, the length of the nucleus and the length of the satellite of the sentence (all of them computed as the total number of words). These features could be indicative of the way in which people write opinions. For instance, a factual sentence could be longer than an opinionated one.

Positional Features. We encode the absolute position of the sentence within the document (e.g., 2 for the second sentence in the document) and its relative position (the absolute position normalised by the number of sentences in a document). We also include the number of sentences in the document as a feature. In this way we can represent if the sentence comes from a short or from a long text. These positional features could be highly indicative of opinions. For instance, writers tend to express their thoughts about the topic of the document at the end.

Table 2 summarises the sentence features considered in our study.

3 Experimental Design

To test our methods we need a collection of labelled sentences. We work here with the English version of the NTCIR-7 English MOAT Research collection [7]. This collection contains news from different sources and provides topics[3], documents that were assessed as relevant to the topics, and annotated data at sentence level. The annotations include both relevance and subjectivity information about the sentences, as well as the identification of the opinion holders. The labels were produced by three different assessors. We constructed our ground truth for subjectivity classification using a majority rule: a sentence is classified as subjective (resp. objective) if at least two assessors tagged it as subjective (resp. objective). As a consequence of this process, our ground truth is composed of 3584 sentences: 2697 sentences judged as objective and 887 judged as subjective. The preprocessing of this collection[4] led to a set of 2218 unigrams and 2812 bigrams.

Baseline. We measured the relative merits of our classification approach against a competitive baseline, OpinionFinder (OF) [3]. OF is a robust subjectivity classifier [8] that is powered by Naive Bayes classifiers trained using linguistic features. Basically, it uses linguistic patterns that are correlated with objectivity (resp. subjectivity) and then using them as features in a machine learning

[3] Textual representations of user needs. The information provided include title and narrative statements.

[4] We did not apply stemming and we did not remove stop words.

Table 1. RST relation types taken into account

Relation	Description
attribution	Clauses containing reporting verbs or cognitive predicates related to reported messages presented in nuclei.
background	Information helping a reader to sufficiently comprehend matters presented in nuclei.
cause	An event leading to a result presented in nuclei.
comparison	Clauses presenting matters which are examined along with matters presented in nuclei in order to establish similarities and dissimilarities.
condition	Hypothetical, future, or otherwise unrealized situations, the realization of which influences the realization of nucleus matters.
contrast	Situations juxtaposed to situations in nuclei, where juxtaposed situations are considered as the same in many respects, yet differing in a few respects, and compared with respect to one or more differences.
elaboration	Rhetorical elements containing additional detail about matters presented in nuclei.
enablement	Rhetorical elements containing information increasing a readers' potential ability of performing actions presented in nuclei.
evaluation	An evaluative comment about the situation presented in the associated nucleus.
explanation	Justifications or reasons for situations presented in nuclei.
joint	No specific relation is assumed to hold with the matters presented in the associated nucleus.
temporal	Clauses describing events with a specific ordering in time with respect to events described in nuclei.

algorithm. Extraction patterns were created by applying a set of syntactic templates to the corpus. These patterns reflect syntactic relationships identified by a shallow parser [9]. Two classifiers are supported by OF: an accuracy classifier and a precision classifier. The first one yields the highest overall accuracy. It tags each sentence as either subjective or objective. The second classifier optimises precision at the expense of recall. We used the first classifier and we adopted the $F1$ score (computed with respect to the subjective class) to evaluate opinion detection effectiveness.

Classification Method. In our experiments we used *liblinear* [10], which is a highly effective library for large-scale linear classification. This library handles Support Vector Machines (SVMs) classification and Logistic Regression classification with different regularisation and loss functions. These classifiers have shown to be very effective in many scenarios. We extensively tested all the classifiers supported by *liblinear* against the training collection to understand what classifier performs the best.

Training & Test. We randomly split the dataset into a training and test set, consisting of 75% and 25% of the sentences, respectively[5]. With the training set

[5] We repeated this process 10 times and we averaged out the performance achieved to obtain a reliable estimation of effectiveness.

Table 2. List of sentence features

Group	Feature
vocabulary	unigrams and bigrams.
Opinion	number of opinionated terms (pos. & neg.). number of exclamations and interrogations. number of opinionated terms (pos. & neg.) normalized by the length of the sentence. number of exclamations and interrogations normalized by the length of the sentence.
RST	Contains a satellite (binary feature). Contains a specific type of satellite: For each relation type reported in Table 1, we represent whether or not the sentence contains that relation type (one binary feature for each type of RST relation).
Position	Number of sentences in the document. Relative position of the sentence in the document. Absolute position of the sentence in the document.
Length	Length of the sentence. Length of the nucleus. Length of the satellite.

we applied 5-fold cross validation to set all the parameters of the classifiers and also to select the best performing classifier[6]. Observe that this two-class classification process is unbalanced: only 887 out of 3584 sentences are labelled as subjective. When dealing with unbalanced problems, discriminative algorithms such as SVMs or Linear Regression, which maximise classification accuracy, result in trivial classifiers that completely ignore the minority class [11]. Some of the typical methods to deal with this problem include oversampling the minority class (by repeating minority examples), under-sampling the majority class (by removing some examples from the majority class), or adjusting the misclassification costs. Oversampling the minority class results in considerable computational costs during training because it significantly increases the size of the training collection. Under-sampling the majority class is not an option for our problem because we have a small number of positive examples and we would need to remove most of the negative examples in order to have sets of positive examples and negative examples that are comparable in size. This massive removal of negative examples would result in much information being missed. We therefore opted for adjusting the misclassification costs to penalise the error of classifying a positive example as negative (i.e., subjective sentence classified as a non-subjective). The training process was designed to maximize the $F1$ score

[6] Usually, the best classifier was a Logistic Regression classifier. The optimal value of the misclassification cost of the subjectivity class was often around 10. The optimal values for C were in the range (0,100].

computed with respect to the subjective class. Next, we used the test set to evaluate the best performing classifier against unseen data.

4 Results

The test results are reported in Table 3. We experimented with different sets of features combined with unigrams and unigrams/bigrams representations. We also include OF's results against the same collection.

Table 3. Experimental results against the test dataset. The best performance for each column and for each set of features is bolded.

	Precision	Recall	F1
OpinionFinder	.4420	.4126	.4268
unigrams	.4926	.3855	.4325
+ Rhetorical	.4903	.4140	.4489
+ Positional	.4716	**.5033**	**.4869**
+ Length	.4571	.4846	.4704
+ Sentiment Lexicon	**.5077**	.4513	.4778
+ All	.4892	.4822	.4857
unigrams & bigrams	.5410	.3591	.4317
+ Rhetorical	.4903	.3576	.4248
+ Positional	.5045	.4573	.4797
+ Length	.4806	.4464	.4629
+ Sentiment Lexicon	**.5517**	.3883	.4558
+ All	.4858	**.5150**	**.5000**

Most of our methods outperform OF, being the unigrams/bigrams representation combined with all other features the best method ($F1$ score of 0.5). Analysing individually the sets of features we can observe that positional features seem to be highly indicative of subjectivity. The potential benefits of positional information to detect subjective information has been also shown in polarity estimation in blogs [12]. Features based on sentiment lexicon and length also contribute to improve the basic classifiers. In contrast, rhetorical information alone modestly improves performance. This does not mean that RST is not a good indicator of subjectivity. In fact, some length features take advantage of RST information (i.e., length of the nuclei/satellite). Moreover, we conducted a preliminary analysis and found that some relations are highly indicative of subjectivity. For example, *attribution* recurrently appears in subjective sentences. In 30% of the subjective sentences this relationship occurs, whereas only 15% of the objective sentences contain an attribution relationship. Another important factor to take into account is that the presence of some RST relations is marginal. For instance, there are only 12 sentences with the *comparison* relation

in our collection. This makes that some RST features are insignificant to discriminate between subjective and objective sentences. However, the inclusion of RST features seems to work well in combination with other evidence (e.g., combined with opinion lexicon features). This indicates that RST can modulate the influence of lexicon-based information. For instance, the presence of a *contrast* satellite could be valuable to increase the subjectivity score of a sentence. Consider the subjective sentence *"A degree of selfishness in capitalist countries seems to be part of the ideology, but one of the great lessons of this bloody 20th century was that pure self-interest needs to be tempered by a contribution to the more general good"*. This sentence has a *contrast* relationship in which the author compares the statements presented in the satellite (*"A degree ... ideology"*) and its nucleus (*"but one ... good"*). This is important evidence in favour of subjectivity (irrespective of the number of opinion terms in the sentence). On the other hand, the presence of a temporal satellite could be evidence in favour of objectivity. For example, the sentence *"Pakistan detonated a series of nuclear devices last month after India surprised the world with its tests"* has a *temporal* relationship between the nucleus (*"Pakistan ... month"*) and the satellite (*"after India ... tests"*). The *temporal* information provides the time when Pakistan detonated the nuclear devices, but the sentence does not express any opinion about it. Observe that with sentiment lexicon features alone, this sentence has a high probability of being classified as subjective because of the presence of opinionated words such as *surprised* or *detonated*. Note also that lexicon-based features lead to high-precision classifiers but the levels of recall are inferior to those achieved by other combination of features.

4.1 Feature Weights

After obtaining a linear SVM model, the weights (w_i) of the separating hyperplane can be used to assess the relevance of each feature [13]. The larger $|w_i|$ is, the more important the i_{th} is in the decision function of the SVM. Only linear SVM models have this indication, which naturally facilitates the analysis of the classifiers. This useful property has been used to gain knowledge of data and, for instance, to do feature selection [13,14]. A proper and direct comparison of the weights can only be done if all features are scaled into the same range. We therefore focus our analysis of feature weights on the *unigrams & bigrams + All* classifier after scaling the features into [0,1]. Table 4 presents the top 50 features ranked by decreasing absolute weight ($|w_i|$). A positive weight ($w_i > 0$) means that high values of the feature are indicative of the membership of the sentence into the subjective class. On the other hand, a negative weight ($w_i < 0$) means that high values of the feature are indicative of the membership of the sentence into the objective class. The two most discriminative features are the number of negative words and the position of the sentence in the document. Remember that the sentence position feature represents the order of a concrete sentence in its document (e.g., the third sentence of a document has a score of 3). A high w_i score makes that the final sentences have more chance of being labelled as subjective when compared to the initial sentences. This seems to indicate that

writers tend to summarise their overall viewpoints at the end of the document. The most discriminative vocabulary features are the unigrams *objections* and *expressed*. The presence of these words in a sentence is a natural indicator of the opinionated nature of the sentence. Another interesting finding is that some of the most discriminative vocabulary features of the subjective class (i.e., unigrams or bigrams) are personal pronouns (e.g., they, I). These pronouns often appear in opinionated sentences to identify the holder of the opinion (e.g., *They are saying that the new product of company X is quite expensive*). On the other hand, the number of exclamations and interrogations has a high negative w_i score. This means that having interrogations or exclamations is indicative of objectivity in this dataset. This is intringuing, because the use of exclamations or interrogations have been associated to subjective content in the literature [4]. This outcome might be related with the nature of this repository (news articles). The use of interrogation or exclamations by journalists could be related to the way of writing to attract people's attention (e.g., open questions). This, however, requires further study and confirmation with other datasets.

If we focus our attention on the ranking of non-vocabulary features (i.e., all features but unigrams or bigrams), reported in Table 5, we observe that there are several RST features that help to detect subjective sentences. The most discriminative features tend to be terms provided by OF lexicon (i.e., Opinion feature set). This exemplifies the discriminative power of an opinion-based lexicon for classification purposes. Still, there are other interesting sets of features that help to understand the overall nature of the sentences. For instance, as we explained before, *evaluation*, *attribution* and *comparison* relationships are highly indicative of opinionated sentences (positive weights). For instance, it is common to find *attribution* statements when the author of the article writes about opinions of other people (e.g., *"According to the new CEO, the future of the company is brilliant"*). On the other hand *temporal* and *background* relationships seem to be indicative of objective sentences (negative weights). *background* statements help to comprehend the matter present in the nucleus. For instance, in the sentence, *"Culturally they are divided into peranakan and totok"*, the *background* satellite (*"Culturally"*) indicates the nature of the information presented in nucleus (*"they are divided into peranakan and totok"*). Additionally, *temporal* statements tend to be objective and are often used to locate events in time. For instance, in the sentence *"The day after the attacks, we saw immediate cancellations"*, the *temporal* satellite (*"The day after the attacks"*) indicates the precise period of time in which the action of the nucleus (*"we saw immediate cancellations"*) is performed.

The results presented here are promising but the overall performance is still modest[7]. The nature of this collection (news articles), in which the authors (journalists) use an informative way of writing is challenging for sentiment detection algorithms. In other scenarios (e.g., blogs or opinion websites), the opinions are more explicit and this facilitates subjectivity classification. However, our results show that positional and discourse features are promising for developing new

[7] A random classifier for our imbalanced problem would get a $F1$ score around 33%.

Table 4. List of the 50 features with the highest $|w_i|$ in the best classifier(scaled). The features are ranked by decreasing $|w_i|$.

rank	w_i	feature	feature set	rank	w_i	feature	feature set
1	3.0439	#Neg	Opinion	26	-1.6449	to use	vocab.
2	2.4448	nSent	Position	27	1.6324	said in	vocab.
3	-2.4210	#ExcInt	Opinion	28	1.6182	programs	vocab.
4	2.3093	objections	vocab.	29	1.6095	ministers	vocab.
5	2.2380	expressed	vocab.	30	1.6087	US economy	vocab.
6	2.2355	they are	vocab.	31	1.5764	#Pos	Opinion
7	-2.2031	nSentsDoc	Length	32	1.5679	oil	vocab.
8	2.1838	globalisation	vocab.	33	1.5389	poor	vocab.
9	2.1239	actions	vocab.	34	1.5289	observers	vocab.
10	2.0839	Nor	vocab.	35	-1.5199	When	vocab.
11	2.0037	notably	vocab.	36	-1.5158	closer	vocab.
12	-1.9996	weather	vocab.	37	1.5126	terrorism	vocab.
13	1.9034	means	vocab.	38	1.5081	to have	vocab.
14	1.8829	something	vocab.	39	1.4957	leadership	vocab.
15	1.8137	I	vocab.	40	1.4938	looks	vocab.
16	-1.8026	market	vocab.	41	1.4859	#NegNorm	Opinion
17	-1.7575	expected	vocab.	42	-1.4843	see	vocab.
18	-1.7527	key	vocab.	43	1.4766	The economic	vocab.
19	-1.7205	will have	vocab.	44	-1.4742	with a	vocab.
20	1.7190	America	vocab.	45	1.4722	sufficient	vocab.
21	1.7002	#PosNorm	Opinion	46	1.4571	has a	vocab.
22	1.6894	should	vocab.	47	1.4555	mother	vocab.
23	1.6823	investors	vocab.	48	-1.4499	may be	vocab.
24	-1.6593	financial	vocab.	49	-1.4496	external	vocab.
25	-1.6522	world economy	vocab.	50	-1.4472	said Mr	vocab.

opinion classifiers able to overcome the limitations of classical vocabulary-based techniques. In fact, our techniques outperform content-based methods and popular opinion classifiers such as OF.

5 Related Work

Searching for relevant opinions within documents is a difficult task [15,8]. In [16] the authors considered the use of the first and the last sentences of a film review and evaluated their effect on accuracy. The impact of term positions in polarity classifiers was also studied in [17]. The results did not substantially differ with those obtained with no positional information. In this paper we have demonstrated that the use of positional information can lead to a better estimation of the subjectivity of the sentences in a news dataset.

Zirn et. al. [18] presented an automatic framework for fine-grained sentiment analysis at sub-sentence level in a product review scenario. Concretely, they used Markov logic to integrate polarity scores from different sentiment lexicons

Table 5. List of the non-vocabulary features with the highest $|w_i|$ in the best(scaled) classifier. The features are ranked by decreasing w_i.

rank	w_i	feature	feature set
1	3.0439	#Neg	Opinion
2	2.4448	nSent	Position
3	-2.4210	#ExcInt	Opinion
4	-2.2031	nSentsDoc	Length
5	1.7002	#PosNorm	Opinion
6	1.5764	#Pos	Opinion
7	1.4859	#NegNorm	Opinion
8	-1.4224	#ExcIntNorm	Opinion
9	1.3025	has *Evaluation* satellite	RST
10	-1.2566	nSentNorm	Position
11	0.9867	has *Attribution* satellite	RST
12	-0.8718	has *Temporal* satellite	RST
13	-0.8442	has *Background* satellite	RST
14	0.4591	has *Comparison* satellite	RST
15	0.4220	lengthSat	Length
16	-0.3927	has *Manner* satellite	RST
17	-0.3338	has *Cause* satellite	RST
18	-0.3034	lengthNuc	Length
19	-0.2612	has *Contrast* satellite	RST
20	0.2319	has *Condition* satellite	RST
21	-0.1997	has *Enablement* satellite	RST
22	0.1643	lengthSent	Length
23	-0.1635	has *Explanation* satellite	RST
24	-0.1170	has *Elaboration* satellite	RST
25	0.1112	has *Joint* satellite	RST
26	-0.0924	hasSat	RST

with information about relations between neighbouring segments of texts. They demonstrated that the use of rhetorical features improves the accuracy of polarity predictions. Somasundaran et al. [19] demonstrated the importance of general discourse analysis in polarity classification of multi-party meetings. The importance of RST for the classification of ambiguous sentences (i.e., sentences with conflicting opinions) was studied in [20]. In [21], the authors worked with film reviews and used RST to determine the importance of terms for polarity classification. With a sentence-level RST-analysis, they were able to outperform a document-level approach based on sentiment lexicons. However, they did not investigate the combination of RST and positional information and their solution works for a coarse-grained problem (document-level polarity estimation). In our paper we have demonstrated that RST can be applied at sentence level in combination with positional and other content-based features and this helps to select key subjective extracts from formal texts (news articles).

6 Conclusions

In this paper we explored the importance of sentence features such as positional or rhetorical features in fine-grained subjectivity classification processes. We demonstrated that these features are valuable and can be combined with more classical methods based on unigrams, bigrams and subjectivity lexicons. In the near future we would like to validate these findings against other datasets and study more advanced ways to combine features and classifiers.

Acknowledgments. This work was funded by *Secretaría de Estado de Investigación, Desarrollo e Innovación* from the Spanish Government under project TIN2012-33867.

References

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2007)
2. Liu, B.: Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers (2012)
3. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proc. of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP 2005 (2005)
4. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the Workshop on Languages in Social Media, LSM 2011, pp. 30–38. Association for Computational Linguistics, Stroudsburg (2011)
5. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text 8(3), 243–281 (1988)
6. Carlson, L., Marcu, D., Okurowski, M.E.: Building a discourse-tagged corpus in the framework of rhetorical structure theory. In: Proceedings of the Second SIGdial Workshop on Discourse and Dialogue, SIGDIAL 2001, vol. 16, pp. 1–10. Association for Computational Linguistics, Stroudsburg (2001)
7. Seki, Y., Evans, D.K., Ku, L.W., Sun, L., Chen, H.H., Kando, N.: Overview of multilingual opinion analysis task at NTCIR-7. In: Proceedings of NTCIR-7 (2008)
8. Santos, R.L.T., He, B., Macdonald, C., Ounis, I.: Integrating proximity to subjective sentences for blog opinion retrieval. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 325–336. Springer, Heidelberg (2009)
9. Wiebe, J., Riloff, E.: Creating subjective and objective sentence classifiers from unannotated texts. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 486–497. Springer, Heidelberg (2005)
10. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874 (2008)
11. Nallapati, R.: Discriminative models for information retrieval. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2004, pp. 64–71. ACM, New York (2004)

12. Chenlo, J.M., Losada, D.E.: Effective and efficient polarity estimation in blogs based on sentence-level evidence. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 365–374. ACM, New York (2011)
13. Chang, Y.W., Lin, C.J.: Feature ranking using linear svm. Journal of Machine Learning Research - Proceedings Track 3, 53–64 (2008)
14. Brank, J., Grobelnik, M., Milić-frayling, N., Mladenić, D.: Feature selection using support vector machines. In: Proc. of the 3rd Int. Conf. on Data Mining Methods and Databases for Engineering, Finance, and Other Fields, pp. 84–89 (2002)
15. Gerani, S., Carman, M.J., Crestani, F.: Proximity-based opinion retrieval. In: Proc. 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 403–410. ACM, New York (2010)
16. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Pr. of the ACL, pp. 271–278 (2004)
17. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? sentiment classification using machine learning techniques. In: Pr. of the Conference on Empirical Methods in Natural Language Processing (2002)
18. Zirn, C., Niepert, M., Stuckenschmidt, H., Strube, M.: Fine-grained sentiment analysis with structural features, vol. (12). Asian Federation of Natural Language Processing (2011)
19. Somasundaran, S., Namata, G., Wiebe, J., Getoor, L.: Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In: Proc. 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, vol. 1, pp. 170–179. ACL, Stroudsburg (2009)
20. Zhou, L., Li, B., Gao, W., Wei, Z., Wong, K.F.: Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In: Proc. Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 162–171. ACL, Stroudsburg (2011)
21. Heerschop, B., Goossen, F., Hogenboom, A., Frasincar, F., Kaymak, U., de Jong, F.: Polarity analysis of texts using discourse structure. In: Proc. 20th ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 1061–1070. ACM Press (2011)

Recall-Oriented Evaluation for Information Retrieval Systems

Bissan Audeh, Philippe Beaune, and Michel Beigbeder

École des Mines de Saint-Étienne,
158 Cours Fauriel, 42023 Saint-Étienne Cedex 2, France
{audeh,beaune,mbeig}@emse.fr

Abstract. In a recall context, the user is interested in retrieving all relevant documents rather than retrieving a few that are at the top of the results list. In this article we propose ROM (Recall Oriented Measure) which takes into account the main elements that should be considered in evaluating information retrieval systems while ordering them in a way explicitly adapted to a recall context.

Keywords: Recall-Oriented Information Retrieval, Evaluation Metrics.

1 Introduction

Finding suitable evaluation metrics for a specific context in information retrieval should depend on a clear identification of users' needs. In a recall-oriented context, where the user is interested in retrieving all relevant documents, a good evaluation metric should be able to promote systems that retrieve a maximum number of relevant documents and privilege a system that retrieves relevant documents earlier in the list of results over another system that retrieves the same number of relevant documents. Many studies [5,2,1,4] have already pointed out the difficulty of measuring the recall, and the problems of using traditional measures in a recall context. These measures are either highly influenced by relevant documents retrieved early in the list, or they ignore the order in which relevant documents are retrieved. One recent measure that resolved an important part of these problems is *PRES* [2]. Although *PRES* doesn't prefer finding relevant documents early in the list over finding a lot of relevant documents, it does penalize retrieving documents at late ranks. This behavior is produced because *PRES* depends on measuring surfaces on the graph Rank/Recall: a small surface could be generated by a weak recall or by relevant documents retrieved at high ranks. For this reason, *PRES* doesn't allow to control what exactly we want to measure. According to the authors, as long as the user finds relevant documents he will continue to check the list of results hoping to find even more relevant documents, which signifies more user effort, and thus justifies the behavior of PRES in penalizing high ranks of relevant documents. In our opinion, supposing that the user prefers finding less relevant documents rather than finding more somewhere far in the results list is a questionable hypothesis in recall-oriented context.

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 29–32, 2013.

2 ROM : The Recall Oriented Measure

ROM is built over the same graph used by PRES and RNorm[3] but with replacing recall by the ranks of relevant documents retrieved in the vertical axe. It is based on the following notions (cf. Fig. 1):

- The height (h): The number of relevant documents retrieved before N_{max}.
- The width (w): The rank of last relevant document retrieved before N_{max}.
- The average precision at N_{max} (AP).

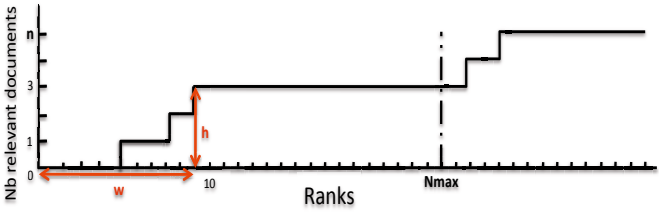


Fig. 1. Height and width notions according to *ROM*

In a recall oriented context, the height should be the most significant part of the measure. To take into account the precision, using only the width is not sufficient, because the measure will not be able to consider the rank of relevant documents retrieved in order to distinguish systems that have the same width. On the other hand, using the average precision alone instead of w will take us back to the problem of getting strongly influenced by the relevant documents retrieved early in the list. *ROM* is represented by a function $f(h, w, AP)$ that takes its values in [0,1]. According to our assumptions regarding user needs in a recall context, this function should have the following properties ($\forall h, h', w, w', AP, AP'$):

p1. $h > h' \Rightarrow f(h, w, AP) > f(h', w', AP')$
p2. $h = h, w < w' \Rightarrow f(h, w, AP) > f(h', w', AP')$
p3. $h = h, w = w', AP > AP' \Rightarrow f(h, w, AP) > f(h', w', AP')$

To fill p1, we consider the sum of h and a function of w and AP that takes its values in [0,1]. Thus our function will have the following representation :

$$f(h, w, AP) = \begin{cases} \frac{h+g_h(w,AP)}{\min(n,N_{max})+1} & \text{if } h > 0 \\ 0 & \text{if } h = 0 \end{cases} \quad (1)$$

where $g_h(w, AP)$ is a function of w and AP for a given h. Following the same logic, we can define the function $g_h(w, AP)$ while reversing the effect of w in order to keep an increasing function. The following definition of g_h thus fills the property (p2):

$$g_h(w, Ap) = \frac{(N_{max} - w) + g_{h,w}(AP)}{(N_{max} - h) + 1} \quad (2)$$

Although AP is already in the range $[0, 1]$, for a given h and w it doesn't necessarily give 0 for the worst scenario nor 1 for the best. That is why we define the function $g_{h,w}(AP)$ (3). Notice that when $h = w$ the best-case, worst-case and actual-case scenarios are identical, thus the function returns the actual AP in this case.

$$g_{h,w}(Ap) = \begin{cases} \frac{AP-AP0}{AP1-AP0} & \text{if } AP0 \neq AP1 \\ AP & \text{if } AP0 = AP1 \end{cases} \tag{3}$$

Where $AP0$ and $AP1$ are respectively the worst and best average precision for a given h and w, they can be obtained by (4).

$$AP0 = \frac{1}{n} \cdot \sum_{i=1}^{h} \frac{i}{w-h+i} \qquad AP1 = \frac{1}{n} \cdot (h - 1 + \frac{h}{w}) \tag{4}$$

Finally, by gathering up (1), (2) and(3) we obtain the final equation of ROM:

$$ROM = \begin{cases} \frac{h(N_{max}-h+1)+N_{max}-w+\frac{AP-AP0}{AP1-AP0}}{(\min(n,N_{max})+1)(N_{max}-h+1)} & \text{if } h > 0 \text{ and } w > h \\ \frac{h(N_{max}-h+1)+N_{max}-w+AP}{(\min(n,N_{max})+1)(N_{max}-h+1)} & \text{if } h > 0 \text{ and } w = h \\ 0 & \text{if } h = 0 \end{cases}$$

where N_{max} is the number of retrieved documents, n is the number of relevant documents, h and w are respectively the count and the last rank of relevant documents retrieved before N_{max}, AP is the average precision at N_{max} , $AP0$ and $AP1$ are average precisions for worst and best scenarios calculated by (4).

3 *ROM* in Practice

Table 1 contains most of the demonstrative examples proposed by [2] in addition to system4 that we have added to demonstrate the advantage of ROM over $PRES$. The systems are ordered according to human preference in a recall context. We notice from Table 1 that ROM is the only measure that respects human ordering. Unlike $PRES$, ROM was able to better place System3 which

Table 1. ROM compared to AP, Recall and $PRES$

	Rank of relevant documents	AP	Recall	$PRES$	ROM
System1	{1, 2, 3, 4}	1	1	1	1
System2	{50, 51, 53, 54}	0.0474	1	0.5	0.895
System3	{1, 98, 99, 100}	0.2727	1	0.28	0.801
System4	{1,54}	0.259	0.5	0.37	0.495

retrieved all relevant documents compared to System4 who retrieved only half of them. Notice that ordering systems 2 and 3 is not easy even for a real user, the small difference in *ROM* score for these two systems is coherent with this fact. In addition to demonstrative examples, we evaluate our measure by means of its relation to other metrics, especially the recall which was the main objective of this work. For our test case, we calculated (for N_{max} =1000) *ROM*, Recall, *PRES* and *MAP* for 88 runs of TREC2012 medical track. From Table 2.(a) we

Table 2. Test results of 88 medical track runs for *ROM* Recall *PRES* and *MAP*

(a) Kendall tau correlation coefficient

Measure pairs	τ
ROM↔ Recall	0.619
ROM↔ *PRES*	0.205
ROM↔ *MAP*	0.093
PRES↔ Recall	0.160

(b) Best and Worst runs

Measure	Best (score)	Worst (score)
ROM	83 (0.819)	49 (0.018)
Recall	83 (0.826)	49 (0.012)
PRES	84 (0.734)	49 (0.011)
MAP	30 (0.461)	49 (0.002)

can see that *ROM* is strongly correlated but not identical to recall, which means that considering w and AP had an important effect on the measure. Observing the relation between *ROM* and *MAP* shows that averaging *ROM* over different queries did not make it prefer precision over recall. From Table 2.(b) we notice that all measures agree on the worst run, but unlike *PRES* and *MAP*, *ROM* sees that the best run is the one which has highest recall (run 83).

4 Conclusion

In this work, we have presented *ROM*, an evaluation measure explicitly adapted to recall-oriented information retrieval. To complete the experiences presented in this paper, we would like to evaluate our measure over larger test sets in other recall contexts like patent and legal search. In addition, we are planning to produce user-centered test cases, where the behavior of the measure is compared to user preferences over some retrieval scenarios in a recall-oriented context.

References

1. Magdy, W.: Toward Higher Effectiveness for Recall-Oriented Information Retrieval: A Patent Retrieval Case Study. Ph.D. thesis, Dublin City University (January 2012)
2. Magdy, W., Jones, G.J.F.: PRES: A score metric for evaluating recall-oriented information retrieval applications. In: SIGIR (2010)
3. Rocchio, J.: Performance indices for document retrieval systems. Tech. Rep. ISR-8, Harvard Computation Laboratory (December 1964)
4. Webber, W.E.: Measurement in Information Retrieval Evaluation. Ph.D. thesis, University of Melbourne (September 2010)
5. Zobel, J., Moffat, A., Park, L.A.: Against recall: is it persistence, cardinality, density, coverage, or totality? SIGIR Forum 43(1), 3–8 (2009)

Using 'Search Transitions' to Study Searchers' Investment of Effort: Experiences with Client and Server Side Logging

Nils Pharo and Ragnar Nordlie

Faculty of Social sciences, Oslo and Akershus University College of Applied Sciences
PO Box 4, St. Olavs Plass, 0130 Oslo, Norway
{nils.pharo,ragnar.nordlie}@hioa.no

Abstract. We are investigating the value of using the concept 'search transition' for studying effort invested in information search processes. In this paper we present findings from a comparative study of data collected from client and server side loggings. The purpose is to see what factors of effort can be captured from the two logging methods. The data stems from studies of searchers interaction with an XML information retrieval system. The searchers interaction was simultaneously logged by a screen capturing software and the IR systems logging facility. In order to identify the advantages and disadvantages we have compared the data gathered from a selection of sessions. We believe there is value in identifying the effort investment in a search process, both to evaluate the quality of the search system and to suggest areas of system intervention in the search process, if effort investment can be detected dynamically.

Keywords: Information retrieval, methods, evaluation.

1 Introduction

Numerous studies have been performed on searchers' interaction with IR systems, in non-web systems [1–3], but in particular with the advent of the Web [4–6]. The study reported in this paper has as its point of departure the notion that effort invested in search processes can be investigated in the light of the concept 'search transition' [7]. Search transitions are constructed to take into consideration the mental effort invested by the searcher during a search process. Effort spent during information searching could be invested in learning to use the system, the adaption of specific system functionalities in the searcher's search strategy; the time spent investigating the details of query result lists etc. Search processes can be split into series of transitions, which in turn can be categorized into different types.

Search transitions can be identified and categorized by thorough analysis of information system transaction logs. There are, we believe, significant differences in the type of information that can be gathered from server and client logs, respectively. Hence it will also differ to what degree the log types provide details of effort invested in the search process. In the present paper we try to answer the following question:

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 33–44, 2013.

What signs of effort can server and client side transaction logs reveal in different types of transitions?

The capturing of server and client logs [8] is a common way to gather data for analyzing IR interaction. One direction of research has been quantitatively oriented studies where researchers have performed analysis of server logs that have captured up to 1 billion queries [4] submitted to the IR system. These kinds of studies have revealed many interesting characteristics of searchers' query formulation and reformulation, e.g. that queries are typically quite short; the use of result lists, e.g. searchers' tendency to only look at a very limited set of documents; and the topicality of queries, e.g. that a large share of web queries are related to pornography.

On the other hand, several studies have been designed that use client side logging, where the goal has been to study e.g. the search processes of searchers across several web sites or to perform very detailed analysis of searchers' interaction with a particular information system. Client logs can be collected either by using client navigation software, such as a web browser, or by screen capturing software. Eye-tracking software can also log the searchers' eye movements over the screen [9].

Server logs reflect the complexity of the information system they capture, e.g. whether the system only contains document surrogates or if it also contains the documents themselves. In a typical web search engine the former will be the case, and primarily interaction with document surrogates will be covered. In order to understand what aspects of effort, as it is understood in our definition of search transitions, can optimally be identified from server logs, we have performed an analysis of a selection of server logs collected by the INEX 2008 interactive track [10], which also contains interaction with the documents. We have compared the server log data with data collected on the client side of the interaction, using the Morae screen capturing software [11]. From this comparison we can learn more on the factors that reflect the searchers' investment of mental effort, and which of these factors that can be identified using server logs, and which factors cannot.

1.1 Measuring Effort Invested in Information Searching

The term "effort", which received an early definition by Fenichel [12] as "a set of search variables [including] e.g. number of commands and descriptors [and] connect time", is quite often considered in the more general literature on information seeking behavior, with this or a variety of other, more or less similar definitions. Zippf's "law of least effort" is often invoked to explain users' choice of information channel[13] , which refers to a number of studies who take this perspective. When effort is considered in the more restricted environment of information search behavior, however, it is often relatively vaguely defined. Typically, it is treated as in [14], where, in an investigation of the influence of user experience on search outcomes, effort is considered as one of several "search language use patterns" and defined to consist of "mean number of cycles per topic, mean command frequency per topic, and mean number of documents visited per cycle" without any motivation for this choice of parameters. A number of authors invoke "cognitive effort" as distinct from observable, logged actions in their characterization of search [15]. Cognitive effort is a

concept well known from fields such as psychology and decision theory, but as a parameter of search effort it is often treated with a similar lack of specific definition as the concept of effort in general. Where it is defined the measurement definitions range widely, from "pupil dilation" in an eye-tracking study of search and evaluation behavior [16] to "number of iterations, i.e. queries in a search" [17].

The term *transition*, or parallel expressions such as shifts, state changes etc. is widely used in both the general literature on information seeking and more specifically in studies of information search behavior. It is generally defined in terms of a move from one state to another (or a sequence of such moves). Stages or patterns of stages appear in more and more fine-grained form in models of information seeking behavior from Ellis' and others' early models [18, 19], and are becoming more and more fine-grained, as in Xie [20], where the interest is in shifting patterns between search stages. Such stages may be identified for instance in information seeking mediation, as in [21] where stages are identified as sets of cognitive and operational elements and transitions between stages are identified through vocabulary changes in dialogue. Transitions have been of particular interest to studies of search system interactions, where it has been thought that being able to detect transitions or distinct shifts in interaction would enable the automatic detection of patterns that might engender some kind of machine assistance or inform interface design. Variants of Markov modeling have often been suggested for such modeling, in [22] weaknesses of this approach is discussed, and an alternative modeling approach with Petri nets are suggested. In this paper and many others the transitions themselves are vaguely defined, and this is a persistent problem in the literature.

We believe our suggested concept, *search transition*, can be used to measure effort in the form of number of search transitions and through analyzing the search transition patterns followed by searchers. Each transition represents a combination of factors involving searcher interaction with information items. Factors that represent mental effort invested during IR interaction include query formulation and reformulation, the selection of source and document types, the number of documents and/or other units of information viewed etc. The rationale behind using search transition as a measure of effort is to take into account the cognitive load required by searchers to deal with a variety of such challenges during interaction. Different IR systems facilitate different types of search transitions, e.g. ISI citation indexes exemplifies a complex IR system with many filtering and refinement options whereas the default search options of web search engines offer quite simple interaction options.

2 Method

The search system applied in the study is a java-based retrieval system built within the Daffodil framework [23], which resides on a server at and is maintained by the University of Duisburg. The search system interface (see Figures 1 and 2) is developed for the INEX (Initiative for the Evaluation of XML retrieval) 2008 interactive track [10]. The database consists of approximately 650 000 Wikipedia articles, which

have been indexed on three levels of granularity; as full article, section level, and subsection level.

Searchers were asked to perform two search sessions, to solve one fact-finding task and one research task, each task was formulated as a simulated work task [24]. Searchers were, for each task, asked to assess the relevance of any document (article) or document element (section or subsection) they viewed during the process. All sessions were logged by the IR system, in order to compare the server and client logs a selection of sessions were also screen captured on the client side using Morae. In our comparative analysis we have looked at 8 sessions in detail to compare the advantages and disadvantages of each of the two logging procedures in connection with identifying different expressions of effort during information searching, relating these to explicit search transitions. In addition we have studied the individual transition patterns of two selected sessions in order to study effort invested throughout the sessions.

2.1 Server Logs

Our server logs captures information about the query input, titles of retrieved information units (i.e. articles, sections and subsections), system-suggested terms for alternative query formulation, titles of information units selected from the result list and the articles table of contents, relevance assessments, internal interaction within individual articles and parts of articles and more. All transition types can be captured by the server logs.

All events in the logs are time-stamped, which means that we can trace the sessions in high detail with respect to the order and selection of events. This makes us able to recreate what interface functionalities were used by the searchers.

2.2 Client Logs

The logs captured at the client side contain all actions made by the searcher during the session, including traces of all mouse movements, highlighting of clicks, continuous time recording to the hundredth part of a second, etc. It is possible to record searchers' utterances/talking aloud, but we choose not to do so for this experiment.

2.3 Log Comparison

Our comparative analysis has focused on characteristics in the two log types that reflect searcher effort. Search transition type (see below) has been used as the organizing factor, i.e. for each transition type we have attempted to make explicit what traces of effort can be found using server and client logs respectively.

2.4 Search Transitions

The following list of search transition types were identified through a study of the server logs of the system used in our experiment:

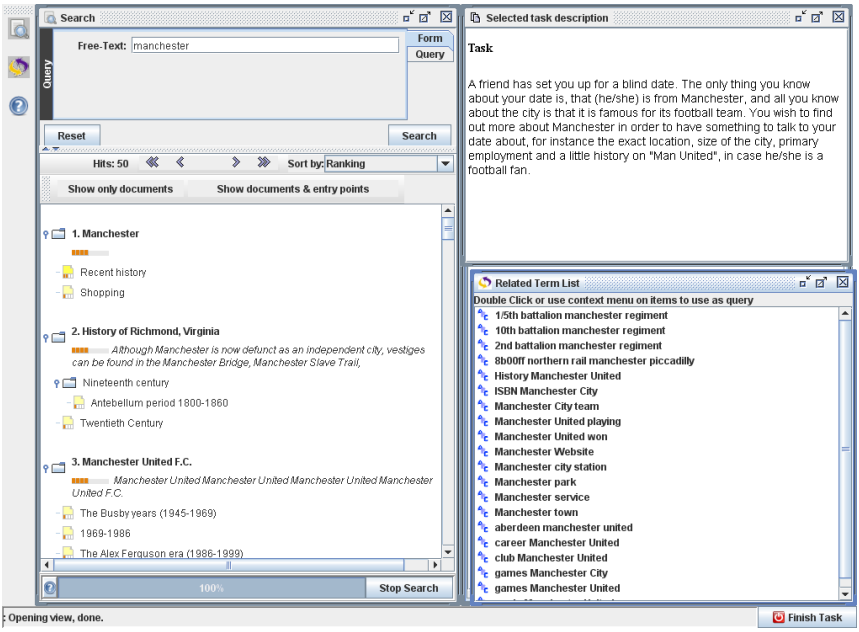


Fig. 1. Search interface of Daffodil

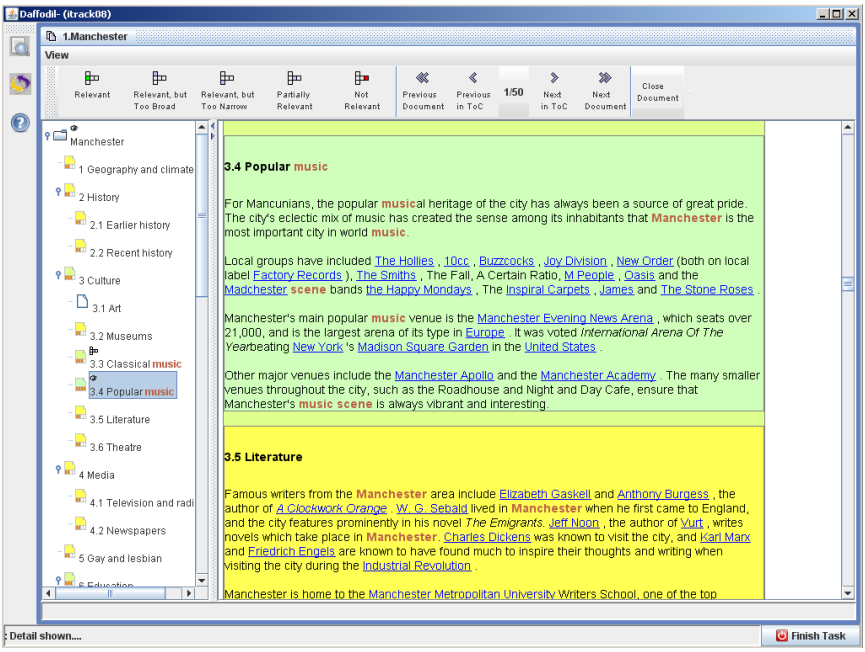


Fig. 2. Document interface of Daffodil

a) Query – result
b) Query – result – inspection
c) Result – inspection
d) Inspection – link to other page – inspection
e) Back button – link to other page – inspection
f) Use system suggested terms – results
g) Use system suggested terms – results – inspection
z) End interaction

Transition a) describes the searcher performing a query in the IR system, but no information unit is selected for further inspection (i.e. selected by a click in the result list, since this is the only expression of "inspection" identifiable in the server log). In transition b) the searcher performs a query, and then from the result list selects a unit of information (a document, a section of a document, or metadata representing the document). In transition type c) the searcher returns to the result list after having inspected a unit of information and then selects a new unit, without a new query being performed. In transition d) the searcher from within an article selects a link to another article. In transition type e) the searcher uses the system's back button to the previous page and then selects a link to another article (note that transition type e) is always preceded by transition type d)). The difference between transition types f) and g) is that in the former the searcher does not select any of the entries in the result list for further inspection after having performed the search on the search term from the suggestion list. Note that interaction within a document, for instance through the TOC, cannot be identified through server logs and is treated as part of the inspection process. Transition z) is used to indicate that system interaction stops, this could be provoked by the searcher logging out of or exiting the system in other ways or by system failure.

3 Findings

The 8 sessions contained a total of 85 server-log-identifiable transitions. Six out of the eight sessions lasted approximately 15 minutes, whereas one lasted seven minutes and 40 seconds and one ended after two minutes and 15 seconds (the latter two were both performed by the same searcher). Table 1 shows the distribution of transition types in the sessions. We see that the large majority of transitions are of types a, b and c, i.e. searchers entering queries, looking at result lists and selecting potentially relevant items. Few examples are found of the use of the system's suggested term feature. No examples of transitions d or e were found in our server log files, the client logs, however, showed many examples of searchers contemplating following internal links. In general the client logs, as expected, revealed many signs of searcher confusion in using the interface.

Table 1. Distribution of transitions in sample

Transition type	
a	25
b	24
c	20
d	0
e	0
f	3
g	2
z	8

We studied the server and client logs in order to find expressions of effort in the different types of transitions. Table 2 summarizes the findings from our log comparison study, and shows *additional* expressions of effort identifiable in the client logs.

Table 2. Server and client log comparison

Transition type	**Server log**	**Client log**
a	Duration in seconds Query terms used Number of items found Titles of the items found	Time spent waiting for system response Time spent contemplating actions (e.g. term selection) Browsing of result lists Reading of text snippets in lists Query reformulations considered, but not executed
b	Same as a) + Number of items looked at (i.e. clicked) Titles of the items looked at Relevance assessments	Same as a) + Browsing of items Reading of items Hesitation in relevance assessments
c	Number of items looked at Titles of the items looked at Relevance assessments	Browsing of items Reading of items Hesitation in relevance assessments
d		
e		
f	Available suggested terms Term(s) selected	Terms considered selected
g	Same as f + Titles of the items looked at	Same as f + Browsing of items Reading of items
z		

We see that the client logs reveal in much more detail how searchers are investing effort in interacting with the text, in particular on how the work load is divided between browsing and reading information items (articles and sections) and browsing metadata, such as titles and related terms (in transition types a, but particularly in type b and g), and on the variation of sources and on the dynamic process of query formulation and reformulation. Server logs, on the other hand, facilitate statistical data analysis due to the logs' capturing of the number of items found and used and the timestamps of all events. We have found, however, examples of server logs mixing up the order of events, making exact capturing of the process in the use of this particular system harder.

Table 3. Two session examples with transitions

Transition #	Session 1	Length	Session 2	Length
1	a	2 min 19 s	a	3 min 43 s
2	a	7 s	c	18 s
3	b	57 s	c	30 s
4	a	1 min 4 s	a	43 s
5	b	1 min 15 s	g	43 s
6	a	1 min 14 s	a	1 min 33 s
7	b	31 s	c	1 min 11 s
8	a	1 min 15 s	a	1 min 33 s
9	c	1 min 8 s	b	1 min 10 s
10	g	54 s	b	1 min 16 s
11	a	58 s	a	1 min 47 s
12	c	40 s	z	0 s
13	c	1 min 39 s		
14	z	0 s		

A close examination of the server log of two sessions showed the distribution of transition types as reported in Table 3. Here transitions are ordered chronologically and we see the time spent on each transition. Both sessions were approximately 15 minutes long.

We see from Table 2 that Type a transitions are the most common in our small sample, but also that the time spent in different transition types differ very much. If we analyze these transactions through the client logs, we find that a lot of the time spent during the session consists of waiting for query results to appear, thus time is not always a good effort indicator. In both sessions the searcher starts with a rather long transition, in Session 1 the searcher spends much time spent in inspecting the query results whereas in Session 2 a large amount of time is spent in inspecting one particular document, also here the searcher hesitates much in deciding whether the document is relevant or not. Also in the first transition in Session 2 the searcher inspect the system's related-term feature, perhaps in order to acquire inspiration for query formulation. No terms are however selected to generate new queries. Transition 10 in Session 2 (a Type b transition) contains an interesting sample of query formulation, here the searcher

spends the approximately first half a minute to formulate two different queries without submitting them before settling for a third version. This kind of effort investment cannot be captured by the server logs. In Session 1, transition 10 (Type g transition), it is interesting to observe that as the searcher is struggling in formulating an effective query with the help of the suggested-term feature, there are several relevant items visible in the result list, but these are overlooked by the searcher. This can perhaps be considered an example of "uni-tasking", i.e. inability to deal with several items in the system's interface (=multi-task) due to heavy effort investment in one particular task. Other examples of time spent during the sessions include waiting for query results to appear, and inspection of result lists to find ideas for query terms.

In Table 4, looking at the first a) + b)-type transition in more detail, we identify the following behavior (based on the premise that mouse movements to a large extent identify the focus of attention on the screen):

Table 4. Two transtitions in detail

00:00	Reads task
00:30	Writes search term (st): Presidential election
00:36	Hesitates
00:41	Extends st: Presidential election France
00:48	Corrects st: Presidential election Europe
00:51	Clicks *search*
00:51	Waits for result (rl) and suggestion (sl) lists
01:14	Inspects rl, finds no relevant item among 3 items shown
01:22	Inspects sl, finds no relevant suggestion
01:29	Changes st; European presidential election
01:35	Clicks *search*
01:35	Waits for result
01:53	Corrects st: Presidential election
01:54	Inspects sl: selects European Election official
01:55	Receives rl for search European presidential election, and continues working with this list

We see that the two identifiable server-side transitions involves several attempts and misunderstandings, waiting time and work time interspersed, all of which is invisible and unanalyzable in the server-side log.

4 Conclusion

We believe there is value in identifying the effort investment in a search process, both to evaluate the quality of the search system and to suggest areas of system intervention in the search process, if effort investment can be detected dynamically. This calls for a machine-identifiable measure of effort, however. Our concept of search transition is

describable and identifiable in server-side logs and should thus be possible to automatically detect and apply. Server-side logs have several advantages:

- It is easy to collect data on time spent on different activities
- Data on query formulation and reformulation can be easily collected for analysis
- Easier countability of events (page retrieval, link selection etc) allows for discovery of general patterns

This is only of value, however, if the effort implied in the server-side transitions are comparable to the effort identified in client-side application of the same definition of transition. The client-side analysis permits, among other things:

- making distinctions between time spent due to hesitation, browsing of content, inspection of interface functionalities, low system response time etc
- Capturing browsing of pages and result lists
- Identification of more details in the query formulation process (e.g. queries that are edited several times before they are submitted)
- Easier understanding of the order of events (server log events are sometimes presented in an un-predictive order)
- Capturing details in the use of system functionalities that are not included in the server logs

Client logs are usable for acquiring valuable data about searchers' effort unavailable from the server. Of importance, for instance, is data that distinguishes between time spent due to system and software problems and time spent by searcher trying to launch his/her search strategies various ways.

Additional data makes it possible to create a more fine-grained taxonomy of search transition types. Different transition types can, e.g. differentiate between the effort spent waiting for system response and the effort invested in creating queries that best match the searcher's current understanding of his/her information need. This fine-gradedness supplements, but does not necessarily replace the transition taxonomy of the server-side logs.

However, in our study of the client logs we have been able to identify several signs of intellectual effort investment in terms of searcher decisions. It may seem that effort, considered in this way, is distributed randomly across different transition types at different stages of the session. Analysis of larger data sets is necessary to identify if there are clear patterns, and whether it is these instances of "micro-effort" rather than the more comprehensive transitions identifiable in the server logs which are best suited to measure user search effort.

We believe that the value gained from analyzing client side logs for understanding more about the mental effort involved from searchers justifies its use. Lessons learnt from usability studies states that from studying a rather small number of users, usability experts are able to identify a large number of the system errors [25]. Is a corresponding pattern to be found when performing microanalysis of client side logs of information search behavior? Our findings at least indicate that the analysis of quite

small sets of data (8 sessions by four different searchers) can be used to identify interesting characteristics of effort investments.

In order to understand more about how searchers invest their mental effort in information searching we suggest that as much data as possible should be collected from the server logs, including timestamps of documents retrieved and accessed from the result lists. Client side logs should complement the analysis of the server logs to identify the specific challenges of the information system in use. Preferably client logs should be collected so that they cover different "environmental" factors as broadly as possible, e.g. search sessions from different times of the day, from different locations, with different client software (e.g. different web browsers) etc. The client log data could then be used to strengthen the understanding of the effort invested in the different search transitions types that are categorized from the server log data.

Acknowledgments. We would like to thank the participants of the INEX 2008 interactive track.

References

1. Ingwersen, P.: Search Procedures in the Library—Analysed from the Cognitive Point of View. J. Doc. 38, 165–191 (1982)
2. Fidel, R.: Moves in online searching. Online Inf. Rev. 9, 61–74 (1985)
3. Bates, M.J.: The design of browsing and berrypicking techniques for the online search interface. Online Inf. Rev. 13, 407–424 (1989)
4. Silverstein, C., Marais, H., Henzinger, M., Moricz, M.: Analysis of a very large web search engine query log. Sigir Forum 33, 6–12 (1999)
5. Spink, A., Jansen, B.J.: Web Search: Public Searching on the Web. Springer (2004)
6. Jansen, B.J., Spink, A.: How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Inf. Process. Manag. 42, 248–263 (2006)
7. Pharo, N., Järvelin, K.: The SST method: a tool for analysing Web information search processes. Inf. Process. Manag. 40, 633–654 (2004)
8. Jansen, B.J., Spink, A., Taksa, I. (eds.): Handbook of Research on Web Log Analysis. IGI Global (2008)
9. Balatsoukas, P., Ruthven, I.: An eye-tracking approach to the analysis of relevance judgments on the Web: The case of Google search engine. J. Am. Soc. Inf. Sci. Technol. 63, 1728–1746 (2012)
10. Pharo, N., Nordlie, R., Fachry, K.N.: Overview of the INEX 2008 Interactive Track. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 300–313. Springer, Heidelberg (2009)
11. Morae usability testing software from TechSmith, `http://www.techsmith.com/morae.html`
12. Fenichel, C.H.: Online searching: Measures that discriminate among users with different types of experiences. J. Am. Soc. Inf. Sci. 32, 23–32 (1981)
13. Bronstein, J., Baruchson-Arbib, S.: The application of cost—benefit and least effort theories in studies of information seeking behavior of humanities scholars: the case of Jewish studies scholars in Israel. J. Inf. Sci. 34, 131–144 (2008)

14. Yuan, W.: End-user searching behavior in information retrieval: A longitudinal study. J. Am. Soc. Inf. Sci. 48, 218–234 (1997)
15. Thatcher, A.: Web search strategies: The influence of Web experience and task type. Inf. Process. Manag. 44, 1308–1329 (2008)
16. Lorigo, L., Pan, B., Hembrooke, H., Joachims, T., Granka, L., Gay, G.: The influence of task and gender on search and evaluation behavior using Google. Inf. Process. Manag. 42, 1123–1131 (2006)
17. Belkin, N.J., Kelly, D., Kim, G., Kim, J.-Y., Lee, H.-J., Muresan, G., Tang, M.-C., Yuan, X.-J., Cool, C.: Query length in interactive information retrieval. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, pp. 205–212. ACM, New York (2003)
18. Belkin, N.J., Cool, C., Stein, A., Thiel, U.: Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. Expert Syst. Appl. 9, 379–395 (1995)
19. Ellis, D.: A behavioural approach to information retrieval system design. J. Doc. 45, 171–212 (1989)
20. Xie, H.: Shifts of interactive intentions and information-seeking strategies in interactive information retrieval. J. Am. Soc. Inf. Sci. 51, 841–857 (2000)
21. Olah, J.: Shifts between search stages during task-performance in mediated information seeking interaction. Proc. Am. Soc. Inf. Sci. Technol. 42, n/a–n/a (2005)
22. Kantor, P.B., Nordlie, R.: Models of the Behavior of People Searching the Internet: A Petri Net Approach. Proc. Asis Annu. Meet. 36, 643–650 (1999)
23. Fuhr, N., Klas, C.-P., Schaefer, A., Mutschke, P.: Daffodil: An Integrated Desktop for Supporting High-Level Search Activities in Federated Digital Libraries. In: Agosti, M., Thanos, C. (eds.) ECDL 2002. LNCS, vol. 2458, pp. 597–612. Springer, Heidelberg (2002)
24. Borlund, P.: Evaluation of interactive information retrieval systems. Abo Akademis Forlag (2000)
25. Nielsen, J., Landauer, T.K.: A mathematical model of the finding of usability problems. In: Proceedings of the INTERACT 1993 and CHI 1993 Conference on Human Factors in Computing Systems, pp. 206–213. ACM, New York (1993)

An IR-Inspired Approach to Recovering Named Entity Tags in Broadcast News

Niraj Shrestha, Ivan Vulić, and Marie-Francine Moens

Department of Computer Science, KU Leuven, Belgium
{niraj.shrestha,ivan.vulic,marie-francine.moens}@cs.kuleuven.be

Abstract. We propose a new approach to improving named entity recognition (NER) in broadcast news speech data. The approach proceeds in two key steps: (1) we automatically detect document alignments between highly similar speech documents and corresponding written news stories that are easily obtainable from the Web; (2) we employ term expansion techniques commonly used in information retrieval to recover named entities that were initially missed by the speech transcriber. We show that our method is able to find named entities missing in the transcribed speech data, and additionally to correct incorrectly assigned named entity tags. Consequently, our novel approach improves state-of-the-art NER results from speech data both in terms of recall and precision.

Keywords: Named entity recognition, term expansion, broadcast news, speech data.

1 Introduction

Named entity recognition (NER) is a task of extracting and classifying information units like *persons*, *locations*, *time*, *dates*, *organization names*, etc. (e.g., [17]). The task involves labeling (proper) nouns with suitable *named entity tags*, and it is usually treated as a sequence prediction problem. NER is an important pre-processing task in many applications in the fields of information retrieval (IR) and natural language processing.

NER in speech data also displays its utility in various multimedia applications. For instance, it could be used in indexing video broadcast news using associated speech data, that is, assigning names and their semantic classes recognized from the speech data as metadata to the video sequences [2]. It is also a useful component of speech-based question answering systems (e.g., [16]), or it could be used to extract names from meeting minutes provided in audio format.

NER in speech data is a difficult task and current state-of-the-art results are typically much lower than the results obtained in written text. For instance, the Stanford NER system in the CoNLL 2003 shared task on NER in written data report an F_1 value of 87.94% [23]. [13,15] report a degrade of NER performance between 20-25% in F_1 value when applying a NER trained on written data to transcribed speech.

This lower performance has several reasons. Firstly, speech transcribers often incorrectly transcribe phrases and even complete sentences, which might consequently result in many missing named entities. Secondly, many names are typically not observed in the training data on which the speech transcriber is trained (e.g., the problem is especially

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 45–57, 2013.

prominent when dealing with dynamic and ever-changing news data). The transcription then results in names and surrounding context words that are spelled incorrectly, making the named entity recognition even more challenging. Finally, the NER system, especially when dealing with such unseen words, might incorrectly recognize and classify the named entities, and even tag non-names with named entity tags.

In this paper, we focus on the first two problems. We assume that similar written documents easily obtainable from the Web discussing the same news events provide additional knowledge about the named entities that are expected to occur in the spoken text. This external knowledge coming from written data then allows finding missing names and correcting incorrectly assigned named entity tags.

We utilize *term expansion and pseudo-relevance feedback techniques* often used in IR. The general idea there is to enrich queries with related terms. These terms are extracted from documents that are selected as being relevant for the query by the user or automatically by the IR system [6]. Only a subset of terms is selected for expansion based on their importance in the relevant document, as well as their semantic relation with the query. We apply a similar approach to expanding and correcting the set of named entities in a speech document by the named entities found in related relevant written documents. Following this modeling intuition, we are able to improve the recall of the NER from broadcast speech data by almost 9%, while precision scores increase for around 0.4% compared to the results of applying the same named entity recognizer on the speech data directly. The main contributions of this article are as follows:

1. We show that NER from speech data benefits from aligning broadcast news data with similar written news data.
2. We present several new methods to recover named entities from speech data by using the external knowledge from high-quality similar written texts.
3. We improve the performance of the state-of-the-art Stanford NER system when applied to the transcribed speech data. The utility of the recovering of missing named entities is especially prominent in much higher recall scores, while we manage to retain a stable and even slightly improved precision level.

The following sections first review prior work, then describe the methodology of our approach and the experimental setup, and finally present our evaluation procedure and discuss the results.

2 Prior Work

There exists a significant body of work on named entity recognition in written data. The task was initially defined in the framework of the Message Understanding Conferences (MUC) [24]. Since then, many conferences and workshops such as the following MUC editions [24,7], the 1999 DARPA broadcast news workshop [21] and the CoNLL shared tasks [22] focused on extending state-of-the-art research on NER.

The most common approach to named entity recognition is based on word-by-word sequential classification techniques, similar to the techniques frequently used for part-of-speech tagging and syntactic base-phrase chunking. A classifier is trained to label each word token in an input text one after the other, in sequence, using the appropriate

named entity tag. Current state-of-the-art NER models typically rely on machine learning algorithms and probabilistic hidden state sequence models such as Hidden Markov Models, Maximum Entropy Markov Models or Conditional Random Fields trained on documents with manually annotated named entities. A myriad of NER implementations are widely available. Examples include the Stanford NER system[1], OpenNLP NameFinder[2], Illinois NER system[3], and LingPipe NER[4]. In this work we utilize the Stanford Named Entity Recognizer, because of its state-of-the-art results, accessible source code and user-friendly interface.

The Stanford NER system [10] identifies named entities of four different types, *person*, *location*, *organization*, and *miscellaneous*.[5] The system recognizes named entities using a combination of three linear chain Conditional Random Field (CRF) sequence taggers. The features used are, among others, word features based on the words in the context window, such as the words themselves and their part-of-speech, orthographic features, prefixes and suffixes of the word to be labeled and surrounding words, and distributional similarity based features. The CRF sequence models are trained on a mixture of various corpora with manually annotated named entities, such as CoNLL, MUC-6 and MUC-7 corpora. These corpora contain both British and American newswire articles, so the resulting models should be fairly robust across domains.

Unfortunately, when applying such a state-of-the-art NER system on transcribed speech data, the performance deteriorates dramatically. In speech data and its transcribed variants, proper names are not capitalized and there are no punctuation marks, while these serve as the key source of evidence for NER in written data. Additionally, speech data might contain incorrectly transcribed words, misspelled words and missing words or chunks of text which makes the NER task even more complex [24,13].

NER in speech data was initiated by [13]. He applied a NER system on transcriptions of broadcast news, and reported that its performance degraded linearly with the word error rate of speech recognition (e.g., missing data, misspelled data and spuriously tagged names). Named entity recognition in speech data has been investigated further, but this related work has focused on either decreasing the error rate when transcribing speech [15,20], on considering different speech transcription hypotheses [11,3], or on the issue of temporal mismatch between training and test data [8]. None of these articles consider exploiting external text sources to improve NER in speech data nor the problem of recovering missing named entities in transcribed speech. Another line of work [5,14] has proven that performing a lexical expansion using related written text obtainable from the Web may boost the performance of systems for speech language modeling, but none of the prior work performed the expansion from written Web sources in the task of NER in speech data. [4,19] link video news stories with written news data. They used closed captions or sub-titles to search related written stories, but do not report on recovering missing named entities.

[1] `http://nlp.stanford.edu/software/stanford-ner-2012-11-11.zip`
[2] `http://opennlp.sourceforge.net/models-1.5`
[3] `http://cogcomp.cs.illinois.edu/page/software_view/4`
[4] `http://alias-i.com/lingpipe/web/models.html`
[5] The system is also able to recognize numerical entities of types *date*, *time*, *money*, and *number*, but we are interested only in the first 3 basic types.

3 Recovering Named Entity Tags in Speech: Methodology

The task is to label a sequence of words $[w_1, w_2, \ldots, w_N]$ from transcribed broadcast news data with a sequence of tags $[t_1, t_2, \ldots, t_N]$, where each word $w_i, i = 1, \ldots, N$, is assigned its corresponding tag t_i. In case of the Stanford NER system utilized in this work, $t_i \in \{person, organization, location\}$.

3.1 Basic Architecture

The straightforward approach to NER in speech data in prior work is to apply a state-of-the-art text data NER tagger (e.g., Stanford NER) directly on transcribed speech data. However, the tagger will miss many named entities or assign incorrect named entity (NE) tags due to the inherent errors in the speech transcription process. In this paper, we use related written text to recover the incorrectly assigned tags and missing named entities in the transcribed speech data. We assume that highly similar written documents or blocks of texts give extra knowledge about the named entities that are incorrectly assigned to the speech data and about the named entities missed in the speech data. The basic modeling work flow follows these steps:

1. Transcribe the speech document using a state-of-the-art ASR system [9] and recognize the named entities in the speech document by a state-of-the-art NER tagger. We call the list of unique named entities obtained in this initial step the *SNERList*.
2. Find related written texts. For instance, news sites often store related written texts with the broadcast video (e.g., Google news). Written news related to the given speech data might also be automatically crawled from the Web. In both cases we use a text similarity metric (e.g., the cosine similarity) to identify related written texts.
3. Group the unique named entities and their tags obtained from the related documents or aligned blocks of written text into the *WNERList*. This list contains valuable knowledge that is utilized to update the *SNERList*.
4. Correct and expand the *SNERList* based on the *WNERList* forming a final list of named entities called FL, the named entities of which can be used as metadata for indexing the speech document. The intuition here is that we should trust the recognized named entities and their tags in the written data more than in the corresponding transcribed speech data.

The models that we propose below differ in the manner they build the complete *SNERList* for a given speech document (Step 4) based on the knowledge in the *WNERList*.

3.2 Baseline NER Model

As a baseline model, we use the Stanford NER system applied on transcribed speech data without any additional knowledge coming from similar written data. We call this model **Baseline NER**.

3.3 Correction and Expansion of the SNERList: General Principles

The procedure proceeds as follows: Let $(x_i)_{t_j}$ be the occurrence of the word x_i tagged by NE class t_j in the *SNERList* and $(x_i)_{t_k}$ be the occurrence of the same word x_i now tagged by the NE class t_k in the *WNERList*. Here, we assume the *one-sense-per-discourse-principle*, that is, all occurrences of the word x_i in a document may only belong to one NE class. We have to update the recognized named entities in the speech transcripts, i.e., replace $(x_i)_{t_j}$ with $(x_i)_{t_k}$ if it holds:

$$Count\big((x_i)_{t_j}\big) < Count\big((x_i)_{t_k}\big) \tag{1}$$

The counts are computed in the most related written document computed in step 2 of the above procedure. This step regards the *correction* of the *SNERList*. This first model that uses the tags of the *WNERList* to correct the *SNERList* is called **NER+COR**. Additionally, we can expand the *SNERList* with named entities from the *WNERList* that were not present in the original *SNERList*. This step denotes the *expansion* of the *SNERList*, but we need to design a smart strategy of selecting named entities from written text that are suitable for the expansion.

3.4 Correction and Expansion of the SNERList Based on the Edit Distance

The model updates the *SNERList* as follows. First, it scans the speech document and searches for orthographically similar words that are tagged in the similar written blocks of the most related written document computed in steps 2 and 3 of the above procedure. Orthographic similarity is modeled by the *edit distance* [18]. We assume that two words are similar if their edit distance is less than 2. The model is similar to NER+COR, but it additionally utilizes orthographic similarity to link words in the speech data to named entities in the *WNERList* in order to expand the *SNERList*. The model is called **NER+COR+EXP-ED**.

These models assign NE tags only to words in the speech document that have their orthographically similar counterparts in the related written data. Therefore, they are unable to recover information that is missing in the transcribed speech document. Hence we need to design additional methods that further expand the *SNERList* with relevant named entities from the written data that are missing in the transcribed speech document. Below we list several alternative approaches to accomplish this goal.

3.5 Expanding the SNERList with Named Entities from Written News Lead Paragraphs

It is often the case that the most prominent and important information occurs in the first few lines of written news (so-called *headlines* or *lead paragraphs*). Named entities occurring in these lead paragraphs are clearly candidates for the expansion of the *SNERList*. Therefore, we select named entities that occur in the first 100 or 200 words in the most related written news story and enrich the *SNERList* with these named entities. Following that, we integrate the correction and expansion of NE tags as before, i.e., this model is similar to NER+COR+EXP-ED, where the only difference lies in the fact that we now consider the additional expansion of the *SNERList* by the named entities appearing in lead paragraphs. This model is called **NER+COR+EXP-ED-LP**.

3.6 Expanding the SNERList with Frequent Named Entities from Written News

The raw frequency of a NE is also a straightforward indicator of its importance in a written news document. Therefore, named entities are selected for expansion of the *SNERList* if they occur at least M times in the most related written document used to build the *WNERList*. Again, the correction part is integrated according to Eq. (1). We build the *SNERList* in the same manner as with the previous NER+COR+EXP-ED-LP model, the only difference is that we now consider frequent words for the expansion of the *SNERList*. This model is called **NER+COR+EXP-ED-FQ**.

3.7 Expanding the SNERList with Frequently Co-occurring Named Entities from Written News

If a NE in the most related written document co-occurs many times with NEs detected in the original speech document, it is very likely that this NE from the written document is highly descriptive for the speech document and should be taken into account for expansion of the *SNERList*. We have designed three models that exploit the co-occurrence following an IR term expansion approach [6].

We compute a score ($SimScore$) for each NE (w_j) in the *WNERList* by which w_j can be ranked according to its relevance for the speech document represented as the set of NEs of the *SNERList*. This co-occurrence score is then modeled in three variant models. The first two models consider the co-occurrence of the NE s_i in the speech document and w_j in blocks of n consecutive words in the written document. So the written document is divided in x blocks B_l. In the third model the distance between s_i and w_j in the written document is taken into account.

(i) Each entity pair (s_i, w_j) consists of one NE from the *SNERList* and one NE from the *WNERList* that is currently not present in the *SNERList* and which is thus a candidate for expansion.

$$SimScore_1(w_j) = \frac{1}{v} \sum_{s_i \in SNERList} \frac{\sum_{B_l} C(s_i, w_j|B_l)}{\sum_{w_k \in WNERList} \sum_{B_l} tf(w_k, B_l)} \tag{2}$$

where $C(s_i, w_j|B_l)$ is the co-occurrence count of NE s_i from the *SNERList* and NE w_j in the written text. The co-occurrence counts are computed over all blocks. $tf(w_k, B_l)$ is the frequency count of the NE w_k in block B_l. We call this model **NER+COR+EXP-ED-M1**. We average the scores over all s_i of the *SNERList*, where v is the number of NEs in the *SNERList*. In a variant model we have normalized the co-occurrence counts of s_i and w_j in block B_l with the co-occurrence counts of s_i with any w_k in block B_l resulting in a very similar performance.

(ii) The next model tracks the occurrence of each tuple (s_i, s_z, w_j) comprising two named entities from the *SNERList* and one NE w_j not present in the list, but which appears in the *WNERList*. The co-occurrence is modeled as follows:

$$SimScore_2(w_j) = \frac{1}{|\Omega|} \sum_{(s_i, s_z) \epsilon \Omega} \frac{\sum_{B_l} C(s_i, s_z, w_j|B_l)}{\sum_{w_k \in WNERList} \sum_{B_l} tf(w_k, B_l)} \tag{3}$$

Again, $C(s_i, s_z, w_j|B)$ is the co-occurrence count of speech named entities s_i and s_z with NE w_j in the written block B_l. Ω refers to all possible combinations of two NEs taken from the *SNERList*. We call this model **NER+COR+EXP-ED-M2**.

(iii) The co-occurrence count in this model is weighted with the minimum distance between NE s_i from the *S*NERList and NE w_j that is a candidate for expansion. It assumes that words whose relative positions in the written document are close to each other are more related. Therefore, each pair is weighted conditioned on the distance between the entities in a pair. The distance is defined as the number of words between the two NEs. The co-occurrence score is then:

$$SimScore_3(w_j) = \frac{\sum_{s_i \in SNERList} \sum_{B_l} \frac{C(s_i, w_j|B_l)}{minDist(s_i, w_j)}}{\sum_{s_i \in SNERList} \sum_{B_l} C(s_i, w_j|B_l)} \quad (4)$$

where $minDist(s_i, w_j)$ denotes the minimum distance between NEs s_i and w_j. The model is called **NER+COR+EXP-ED-M3**.

These 3 models are similar to the other models that perform the expansion of the *S*NERList. The difference is that the expansion is performed only with candidates from the *WNERList* that frequently co-occur with other named entities from the *S*NERList. The notion of "frequent co-occurrence" is specified by a threshold parameter and only entities that score above the threshold are used for expansion.

3.8 Expanding the SNERList with Intersection between Named Entities from a Set of Related Written News Documents

The idea of this model is to retain only the named entities that occur in many related news documents (computed in step 2 of the above procedure) as candidates for the expansion of the *SNERList*. Here, we first select a set of related written news text for each speech document, and then, to expand the *SNERList*, we use the intersection of the named entities, i.e., named entities that occur in all written documents in the corresponding set. We can select the related written documents based on a minimum similarity value, or based on a cut-off in the ranked list of related written documents. Selecting a minimum similarity value is not straightforward. If we take a very low similarity score, it might introduce one or more unrelated written documents. In that case, we might end up with an empty intersection list. If we choose a high similarity score, we might lose relevant related written documents. This model is named as **NER+COR+EXP-ED-INTS**. The selection of K related documents is easier. We might even choose the K related written stories that can be found on the same event on a given date, where K is a flexible number. In the experiments below we use a fixed number K for all speech examples. This model is called as **NER+COR+EXP-ED-INTS-BK**.

All the above models, some of which are borrowed from query expansion research in IR, show the many possible ways of exploiting the named entities in written documents that are related to the speech document in which we want to improve NER.

4 Experimental Setup

4.1 Datasets and Ground Truth

For evaluation, we have downloaded 40 short broadcast news stories from the Web in the periods of October-November 2012 and April-May 2013 randomly selected from `www.googlenews.com`, `tv.msnbc.com`, `bbc.com`, `cnn.com`, and

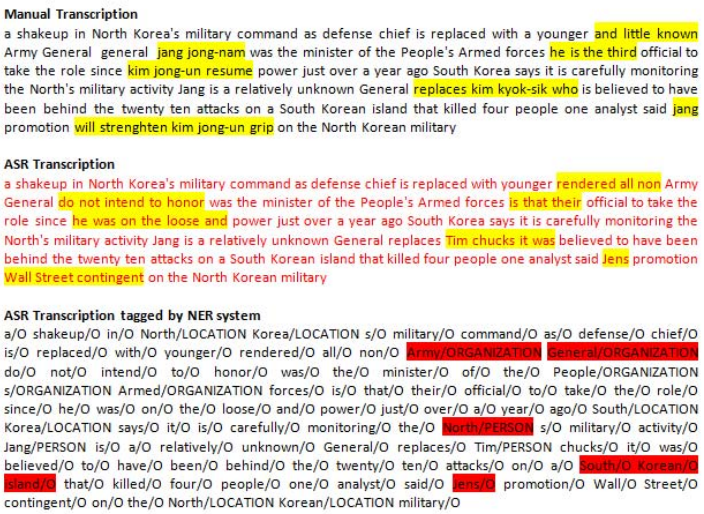


Fig. 1. An example of the actual transcription performed manually, the transcription obtained by the FBK ASR system and the ASR transcription tagged by the Stanford NER system

Table 1. Statistics of 40 broadcast news data used for evaluation

	Frequency of named entities
# **NEs in ground truth**	408
# **NEs transcribed** by FBK ASR	302
# **NEs not transcribed** by FBK ASR (missing names)	106
# **NEs tagged** by Stanford NER	487
# **NEs correctly tagged** by Stanford NER	283
# **NEs incorrectly tagged** by Stanford NER	204

`www.dailymail.co.uk`.[6] We have collected 5532 related news stories from `www.news.google.com` which stores related news stories from different sites, and they constitute our *written text dataset*. The FBK ASR transcription system [9] is used to provide the speech transcriptions of these stories. Since the system takes sound as input, we have extracted the audio files in the mp3 format using the *ffmpeg* tool [1]. The transcribed speech data constitute our *speech dataset*. Fig. 1 shows an example of the manual transcription, and its tagging by the Stanford NER system. It is clear that the ASR transcription contains many words that are incorrectly transcribed and that the ASR system does not recognize many words from the actual speech. It is also noted that the NER system could not tag the missed named entities in the ASR transcription.

In order to build the ground truth for our experiments, all 40 broadcast news stories were manually transcribed. The Stanford NER was then applied on the manually transcribed data. Following that, an annotator checked and revised the tagged named

[6] Dataset is available at `http://people.cs.kuleuven.be/~niraj.shrestha/NER`

Table 2. Results of different NE recovering models on the evaluation dataset

NER Model	Precision	Recall	F_1
Baseline NER	**0.508**	**0.605**	**0.553**
NER+COR	0.521	0.620	0.566
NER+COR+EXP-ED	0.506	0.632	0.562
NER+COR+EXP-ED-LP ($\|LP\| = 100$)	0.444	0.706	0.545
NER+COR+EXP-ED-LP ($\|LP\| = 200$)	0.393	0.718	0.508
NER+COR+EXP-ED-FQ ($M = 2$)	0.438	0.686	0.535
NER+COR+EXP-ED-FQ ($M = 3$)	0.490	0.674	0.568
NER+COR+EXP-ED-M1	0.518	0.634	0.570
NER+COR+EXP-ED-M2	0.516	0.632	0.568
NER+COR+EXP-ED-M3	0.377	0.662	0.480
NER+COR+EXP-ED-INTS-BK ($K = 3$)	0.479	0.725	0.577
NER+COR+EXP-ED-INTS-BK ($K = 5$)	**0.512**	**0.694**	**0.589**
NER+COR+EXP-ED-INTS-BK ($K = 7$)	0.517	0.662	0.581
NER+COR+EXP-ED-INTS-BK ($K = 10$)	0.520	0.642	0.575

entities. The detailed statistics are provided in Table 1. There are all together 106 named entities missing from the speech data set due to transcription errors, and these cannot be tagged by the NER system. Additionally, we observe that a large portion of the named entities is incorrectly tagged by the Stanford NER. The knowledge from aligned written data should help us resolve these issues.

4.2 Evaluation Metrics

Let FL be the final list of named entities with their corresponding tags retrieved by our system for all speech documents, and GL the complete ground truth list. We use standard precision ($Prec$), recall (Rec) and F_1 scores for evaluation:

$$Prec = \frac{|FL \cap GL|}{|FL|} \quad Rec = \frac{|FL \cap GL|}{|GL|} \quad F_1 = 2 \cdot \frac{Prec \cdot Rec}{Prec + Rec}$$

We perform an *evaluation at the document level*, that is, we disregard multiple occurrences of the same named entity in one document. In cases when the same named entity is assigned different tags in the same document (e.g., *Kerry* could be tagged as *person* and as *organization* in the same document), we penalize the system by always treating it as an incorrect entry in the final list FL.

This evaluation is useful when one wants to index a speech document as a whole and considers the recognized named entities and their tags as document metadata. Within this evaluation setting it is also possible to observe the models' ability to recover missed named entities in speech data.

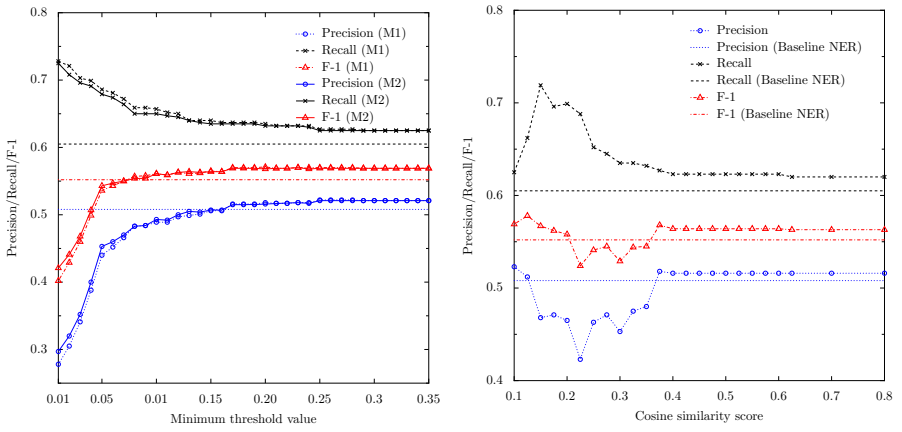
5 Results and Discussion

Table 2 displays all results obtained on the evaluation dataset of 40 broadcast news stories. We compare the results of our models to the baseline model that uses a NER

system directly on transcribed speech data (Baseline NER). We observe that all the proposed models are able to correct a portion of the named entities initially missed in the transcribed speech data (for instance, we notice a small performance boost already by using the simple NER+COR model), and additionally expand the *SNERList* by named entities from similar written data (see performance boosts, especially boosts in terms of recall for all models that perform the *SNERList* expansion). A majority of the proposed models outperform the baseline NER system in terms of F_1 score, and they all exhibit significant performance boosts in terms of recall. The best results are obtained by the NER+COR+EXP-ED-INTS-BK model with $K = 5$, where we can observe an increase of 9% in terms of recall (due to the expansion procedure) (significant for $p < 0.0002$ 2-paired t-test), and a 0.4% increase in terms of precision (significant for $p < 0.14$) that is altogether reflected in a 3.6% increase in terms of F_1 score.

We have investigated the influence of the minimum threshold values on the results obtained by the term co-occurrence models (NER+COR+EXP-ED-M1/M2). Figure 2(a) displays the dependence on the threshold value. We observe that by setting a low threshold we are able to recover a considerable number of named entities (a 12% increase in recall for the threshold of 0.01), but it degrades the precision scores. The best results presented in Table 2 are obtained by the threshold value of 0.2.

We have also investigated the influence of the minimum cosine similarity score that is needed to consider a written document similar to the given speech document in the NER+COR+EXP+ED+INTS model. The results are displayed in Fig. 2(b). If we choose a lower similarity threshold then the model tends to select unrelated written documents as similar to the given speech document and it has a negative impact on the overall results. On the other hand, if the selected threshold is set too high, the model tends to omit relevant related written documents. Results in Table 2 are obtained by setting the similarity threshold to 0.125, but we observe a stable performance for other threshold



(a) (b)

Fig. 2. Influence of parameters on the overall results: (a) threshold value for the term co-occurrence models NER+COR+EXP-ED-M1 and NER+COR+EXP-ED-M2, (b) minimum similarity value for the intersection model NER+COR+EXP-ED-INTS

settings. Similarly, for the K-best cut-off intersection model, we have varied the cut-off position ($K = 3, 5, 7, 10$). The results are displayed in Table 2. Since the F_1 score is stable for different K values, it confirms our hypothesis that K might be chosen in a flexible way, for instance, as the number of written documents on the same event reported in the speech document available on a certain day.

To recover the missing named entities, the system should learn from the related written text. Out of 106 missing named entities (see the statistics in Table 1), there are only 89 named entities observed in the related written news dataset. This constitutes the upper bound of our approach. In order to deal with this problem, we need to collect more related news stories. Furthermore, we have noticed that out of 17 missing named entities, NEs like news anchor or reporter names can rarely be found in related written text. Our best model NER+COR+EXP-ED-INTS-BK with $K = 5$ recovers 31 named entities out of 106 missing named entities from the related news texts boosting the recall substantially without hurting precision.

We are able to recover a substantially larger amount of missing named entities by lowering the threshold for the similarity score computed in Eq. (2) and (3) in models NER+COR+EXP-ED-M1/M2. For instance, as shown in the Fig. 2(a) when we lower the threshold to 0.01, the recall increases to 72.8% and the system recovers 53 missing named entities, but the increased recall is at the expense of a much lower precision ($P = 27.78\%$). In that setting many irrelevant named entities are added to the *SNERList*. Our methods can still be improved by finding better correlations between named entities found in the speech and related written documents. One line of our future research will strive to retain the substantial increases in terms of recall while retaining a stable precision level.

The NE recognition in the related written texts is not perfect either and can entail errors in the correction and expansion of the named entities found in the speech data. [12] report that the performance of the Stanford NER system in Web data decreases by 14%. To confirm this finding, we have also checked the performance of Stanford NER when applied on our written text. We have randomly selected 20 written news stories and run the Stanford NER. The performance is ($P = 76.69\%, R = 80.89\%, F_1 = 78.73\%$) which clearly indicates that there is still ample room for improvement in the task of NER in written data. Further improvements in NER in written data will also have a positive impact on the models for NER in speech data that we propose in this article.

6 Conclusions and Future Work

We have proposed a novel IR-inspired approach to recovering NE tags in transcribed speech using similar written texts. We have shown that NER from speech data benefits from aligning broadcast news data with related written news data. Our new models are able to both (1) correct tags for named entities identified in the speech data that were tagged incorrectly, and (2) expand the list of named entities in the speech data based on the knowledge of named entities from related written news stories. The best improvements in terms of precision and recall of the NER are obtained by considering the named entities that occur in the intersection of several related written documents.

Our results show that we can improve the recall of the NER by 9% compared to solely considering NER in the transcribed speech data without hurting precision. In our evaluation dataset almost 25% of named entities were missing after the ASR transcription, and we have shown that our best method is able to correctly recover and tag almost one third of the missing named entities.

In future work we plan to further refine the NE expansion techniques in order to enrich the lists of named entities in speech using written data without sacrificing precision. We also plan to explore several other speech transcription hypotheses, and study the core problem of domain adaptation when dealing with the task of NE recognition in order to build more portable NER taggers.

References

1. ffmpeg audio/video tool @ONLINE (2012), http://www.ffmpeg.org
2. Basili, R., Cammisa, M., Donati, E.: RitroveRAI: A Web application for semantic indexing and hyperlinking of multimedia news. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 97–111. Springer, Heidelberg (2005)
3. Béchet, F., Gorin, A.L., Wright, J.H., Tur, D.H.: Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How may I help you? Speech Comm. 42(2), 207–225 (2004)
4. Blanco, R., De Francisci Morales, G., Silvestri, F.: Towards leveraging closed captions for news retrieval. In: Proc. of WWW Companion, pp. 135–136 (2013)
5. Bulyko, I., Ostendorf, M., Stolcke, A.: Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures. In: Proc. of NAACL-HLT, pp. 7–9 (2003)
6. Cao, G., Nie, J.Y., Gao, J., Robertson, S.: Selecting good expansion terms for pseudo-relevance feedback. In: Proc. of SIGIR, pp. 243–250 (2008)
7. Chinchor, N.A.: MUC-7 named entity task definition (version 3.5). In: Proc. of MUC (1997)
8. Favre, B., Béchet, F., Nocera, P.: Robust named entity extraction from large spoken archives. In: Proc. of EMNLP, pp. 491–498 (2005)
9. FBK: FBK ASR transcription (2013), https://hlt-tools.fbk.eu/tosca/publish/ASR/transcribe
10. Finkel, J.R., Grenager, T., Manning, C.D.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proc. of ACL, pp. 363–370 (2005)
11. Horlock, J., King, S.: Discriminative methods for improving named entity extraction on speech data. In: Proc. of EUROSPEECH, pp. 2765–2768 (2003)
12. Kim, M.H., Compton, P.: Improving the performance of a named entity recognition system with knowledge acquisition. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 97–113. Springer, Heidelberg (2012)
13. Kubala, F., Schwartz, R., Stone, R., Weischedel, R.: Named entity extraction from speech. In: Proc. of the DARPA Broadcast News Transcription and Understanding, pp. 287–292 (1998)
14. Lei, X., Wang, W., Stolcke, A.: Data-driven lexicon expansion for Mandarin broadcast news and conversation speech recognition. In: Proc. of ICASSP, pp. 4329–4332 (2009)
15. Miller, D., Schwartz, R., Weischedel, R., Stone, R.: Named entity extraction from broadcast news. In: Proc. of the DARPA Broadcast News, pp. 37–40 (1999)
16. Mishra, T., Bangalore, S.: Qme!: A speech-based question-answering system on mobile devices. In: Proc. of NAACL-HLT, pp. 55–63 (2010)

17. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
18. Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys 33(1), 31–88 (2001)
19. Odijk, D., Meij, E., de Rijke, M.: Feeding the second screen: semantic linking based on subtitles. In: Proc. of the 10th Conference on OAIR, OAIR 2013, pp. 9–16 (2013)
20. Palmer, D.D., Ostendorf, M., Burger, J.D.: Robust information extraction from automatically generated speech transcriptions. Speech Comm. 32(1-2), 95–109 (2000)
21. Przybocki, J.M., Fiscus, J.G., Garofolo, J.S., Pallett, D.S.: HUB-4 information extraction evaluation. In: Proc. of the DARPA Broadcast News, pp. 13–18 (1999)
22. Sang, E.F.T.K., Meulder, F.D.: Introduction to the CoNLL-2003 shared task: Language-Independent named entity recognition. In: Proc. of CoNLL, pp. 142–147 (2003)
23. Stanford: Stanford NER in CoNLL 2003 (2003), `http://nlp.stanford.edu/projects/project-ner.shtml`
24. Sundheim, B.: Overview of results of the MUC-6 evaluation. In: Proc. of MUC, pp. 13–31 (1995)

An Exploratory Study on Content-Based Filtering of Call for Papers

Germán Hurtado Martín[1,2], Steven Schockaert[3], Chris Cornelis[2,4], and Helga Naessens[1]

[1] Faculty of Applied Engineering Sciences, University College Ghent, Ghent, Belgium
[2] Dept. of Applied Math., Comp. Science and Statistics, Ghent University, Belgium
[3] School of Computer Science & Informatics, Cardiff University, UK
[4] Dept. of Computer Science and Artificial Intelligence, University of Granada, Spain

Abstract. Due to the increasing number of conferences, researchers need to spend more and more time browsing through the respective calls for papers (CFPs) to identify those conferences which might be of interest to them. In this paper we study several content-based techniques to filter CFPs retrieved from the web. To this end, we explore how to exploit the information available in a typical CFP: a short introductory text, topics in the scope of the conference, and the names of the people in the program committee. While the introductory text and the topics can be directly used to model the document (e.g. to derive a tf-idf weighted vector), the names of the members of the program committee can be used in several indirect ways. One strategy we pursue in particular is to take into account the papers that these people have recently written. Along similar lines, to find out the research interests of the users, and thus to decide which CFPs to select, we look at the abstracts of the papers that they have recently written. We compare and contrast a number of approaches based on the vector space model and on generative language models.

Keywords: Recommendation, Call for papers, Information retrieval, Language models, Vector space model.

1 Introduction

Nowadays many conferences are organized, resulting in a high number of calls for papers (CFPs). This increasing number of CFPs, however, means for the researchers a substantial amount of time spent looking for potentially interesting conferences. The problem has been addressed in several ways, the most popular being the use of domain-specific mailing lists (e.g. DBWorld[1]), or organizing CFPs per subject on dedicated websites (e.g. WikiCFP[2], CFP List[3],

[1] http://research.cs.wisc.edu/dbworld/
[2] http://www.wikicfp.com
[3] http://www.cfplist.com

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 58–69, 2013.

or PapersInvited[4]). However, these solutions still require users to spend part of their time searching for CFPs, and the results do not always match their specific interests.

A number of recent techniques have been proposed for recommending scientific resources such as research papers, with the study and emergence of research paper recommenders [1,5], citation recommendation [9], or applications to find experts in a specific research area [2]. However, to our knowledge, CFP recommendation remains unexplored.

Recommenders typically rely on collaborative filtering approaches [10], content-based methods [6], or hybrid methods. It can be expected that a CFP recommender would be most effective when content-based methods are combined with other techniques. However, before such a recommender can be developed, we feel that a number of content-based aspects need to be understood better, including how the research interests of a user can be induced from his publication history and how these interests could be matched to CFPs. The aim of this paper is to explore which methods may be most suitable for this task. In particular, we consider the textual content of the CFP such as the introductory text or the list of topics, and we complement that information with the abstracts of the papers recently written by the members of the program committee who are named in the CFP. This latter idea has already been used to address the review assignment problem [4,11]. Similarly, we use the abstracts of the papers that the users have previously written to discover their research interests.

The paper is structured as follows. First we discuss in more detail what types of information are at our disposal, and how this information can be used. Subsequently, in Section 3 we introduce different methods to effectively model and compare CFPs and user profiles. In Section 4 we present our experimental results. Finally, in Section 5 we summarize the conclusions of the paper.

2 Available Information

2.1 User Representation

To represent the research interests of users we exploit the papers they have written. Since research interests might change, only recent papers are considered. In our experiments we have considered papers written in the last five years as being recent, although more advanced methods could be envisaged to analyze how the research interests of a user are changing over time. Alternatively, in the case of users with few or no papers (e.g. a beginning researcher) users could specify those papers which represent their interests best. Since getting access to the full text of research papers is not always possible, we only use the papers' abstracts. We then consider, for each user, a document consisting of the concatenation of the abstracts of his papers. For the sake of clarity, we further refer to this document as d_{abs}.

[4] `http://www.papersinvited.com`

What we can also learn from an author's publication profile is which authors he frequently cites. This information can be valuable if we consider that authors are more likely to be interested in conferences whose program committee (PC) contains several people who are working in the same field and whose papers they sometimes cite. To take this into account, we will use a document consisting of the concatenation of the abstracts of the papers written by the authors usually cited by the user. In our experiments, we considered an author to be usually cited if at least 3 different papers written by him have been cited by the user in 3 different works. We refer to this document as d_{aut}.

2.2 CFP Representation

For this work we have used CFPs available from DBWorld. Although there is no standard format for writing CFPs, they usually include similar information: an introductory text about the conference, an indicative list of topics that are within the scope of the conference, and the names of the members of the program committee (or at least the organizers). They usually also include important dates and location, but we will disregard that information.

The introductory text usually consists of a short description about the conference which might contain terms that describe the scope of the conference and are therefore important. However, this description often also refers to past conferences, the proceedings, etc., which means that many terms are mentioned that are not representative of the topics of the conference. We try to compensate this by concatenating this text with the list of topics that are within the scope of the conference. We use the resulting document, which we further refer to as d_{txt}, to model a CFP document.

The names of the members of the program committee are also potentially useful. An option to use them directly could be trying to match them to the names cited in the papers of the users, but the results of initial experiments along these lines were not positive. However, these names can be used indirectly too. In particular, for the experiments reported in this paper, we associate each CFP with a document d_{con}, consisting of the concatenation of the abstracts of all papers that have been written in the last two years by its PC members.

Finally, if we want to consider both types of information simultaneously, we can concatenate d_{txt} and d_{con}; we refer to this document as d_{tot}.

3 Matching CFPs and Users

In this section we review some methods to model and compare users and CFPs, based on the documents defined in the previous section.

3.1 Tf-idf

To measure the similarity between a CFP and a user profile we compare them in the vector space model: each profile is represented as a vector, with one component for every term (unigram) occurring in the collection. A CFP is encoded

as a vector as follows. Stopwords[5] are first removed, no stemming is used. Then, the weight for each term w_i in the CFP profile is calculated by using the tf-idf scoring technique [8]:

$$tfidf(w_i, d_{txt}) = \frac{n(w_i, d_{txt})}{|d_{txt}|} \cdot log(\frac{|\mathcal{C}_{txt}|}{|\{d_j : w_i \in d_j\}|}) \quad (1)$$

where $\mathcal{C}_{txt}$ is the collection of CFPs made from the concatenation of introductory text and scope topics (i.e., of documents of the form d_{txt}).

As mentioned in the previous section, CFP profiles can be represented in different ways. If the documents of the form d_{con} are used instead of those of the form d_{txt}, d_{txt} and $\mathcal{C}_{txt}$ in Eq. (1) are replaced by d_{con} and $\mathcal{C}_{con}$ respectively, where $\mathcal{C}_{con}$ is the collection of CFPs made from the concatenation of the abstracts of the papers written by the PC members (documents of the form d_{con}). On the other hand, if the documents of the form d_{tot} are used, d_{txt} and $\mathcal{C}_{txt}$ in Eq. (1) are replaced by d_{tot} and $\mathcal{C}_{tot}$ respectively, where $\mathcal{C}_{tot}$ is the collection of CFPs made from the concatenation of both textual content and abstracts of the papers written by the PC members (documents of the form d_{tot}).

Since user and CFP profiles belong to different collections, we consider user profiles as queries, and therefore the process to convert a user profile into a vector is slightly different. As with CFP profiles, stopwords are removed and no stemming is used; however, only those terms that occur in the CFP collection are considered, and the rest are ignored. Then the weight of each term in the user profile is calculated by replacing d_{txt} and $\mathcal{C}_{txt}$ in Eq. (1) by d_{abs}^{txt} and $\mathcal{C}_{txt}$, d_{abs}^{con} and $\mathcal{C}_{con}$ or d_{abs}^{tot} and $\mathcal{C}_{tot}$, depending on the type of information used, where d_{abs}^{txt}, d_{abs}^{con} and d_{abs}^{tot} are obtained from the user profile d_{abs} after removing all terms that do not occur in $\mathcal{C}_{txt}$, $\mathcal{C}_{con}$ and $\mathcal{C}_{tot}$ respectively.

Two vectors $\mathbf{d_1}$ and $\mathbf{d_2}$ corresponding to different profiles can then be compared using a standard similarity measure; we use the cosine similarity, defined by

$$sim_c(\mathbf{d_1}, \mathbf{d_2}) = \frac{\mathbf{d_1} \cdot \mathbf{d_2}}{\|d_1\| \cdot \|d_2\|} \quad (2)$$

where $\mathbf{d_1} \cdot \mathbf{d_2}$ denotes the scalar product and $\|.\|$ the Euclidean norm. In Section 4.2 we refer to the method that combines tf-idf with the cosine similarity measure as *tfidf-txt*, *tfidf-con* and *tfidf-tot*, depending on the information used.

3.2 Language Modeling

A different approach is to estimate unigram language models [7] for each document, and determine their divergence. A user profile or CFP d is then assumed to be generated by a given model D. This model is estimated from the terms that occur in d and in the other CFPs from the considered collection. Using

[5] The list of stopwords we have used for the experiments was taken from `http://snowball.tartarus.org/algorithms/english/stop.txt`, expanded with the following extra terms: *almost*, *either*, *without*, and *neither*.

Jelinek-Mercer smoothing, the probability that model D corresponding to a CFP generates term w is estimated as:

$$P^*(w|D) = \lambda P(w|d_{txt}) + (1-\lambda)P(w|\mathcal{C}_{txt}) \tag{3}$$

where $\mathcal{C}_{txt}$ is the collection of CFP profiles as defined in Section 3.1, and the probabilities $P(w|d)$ and $P(w|\mathcal{C})$ are estimated using maximum likelihood, e.g. $P(w|d)$ is the percentage of occurrences of term w in profile d. Alternatively, d_{txt} and $\mathcal{C}_{txt}$ in Eq. (3) can be replaced by d_{con} and $\mathcal{C}_{con}$ if the documents of the form d_{con} are used. To estimate the probability that the model of a user profile generates a given term w we simply replace d_{txt} in (3) by d_{abs}^{txt} or d_{abs}^{con} (as defined in Section 3.1).

Once the models D_1 and D_2 corresponding to a user profile d_1 and a CFP d_2 are estimated, we measure their dissimilarity using the Kullback-Leibler divergence:

$$KLD(D_1||D_2) = \sum_w D_1(w) log \frac{D_1(w)}{D_2(w)} \tag{4}$$

We further refer to these methods as *lm-txt* and *lm-con*.

However, if we want to consider both kinds of information jointly (i.e. the information from the documents of the form d_{txt} and that from the documents of the form d_{con}), language model interpolation is used:

$$P^*(w|D) = \lambda_1 P(w|d_{txt}) + \lambda_2 P(w|d_{con}) + \lambda_3 P(w|\mathcal{C}_{txt}) + \lambda_4 P(w|\mathcal{C}_{con}) \tag{5}$$

with $\sum_i \lambda_i = 1$ and where

$$\lambda_3 = \begin{cases} \frac{1-\lambda_1-\lambda_2}{2}, & \text{if } \lambda_1, \lambda_2 > 0 \\ 1-\lambda_1, & \text{if } \lambda_2 = 0 \\ 1-\lambda_2, & \text{if } \lambda_1 = 0 \end{cases} \qquad \lambda_4 = \begin{cases} \frac{1-\lambda_1-\lambda_2}{2}, & \text{if } \lambda_1, \lambda_2 > 0 \\ 1-\lambda_1, & \text{if } \lambda_2 = 0 \\ 1-\lambda_2, & \text{if } \lambda_1 = 0 \end{cases}$$

To estimate the probability for the user profile, we replace d_{txt} and d_{con} in Eq. (5) by d_{abs}^{txt} and $d_a^{con}bs$. In Section 4.2 we refer to this method as *lm-tot*.

3.3 Feature Selection

As mentioned in Section 2, the introductory texts of the CFPs often contain information about past editions of the conference or brief submission guidelines. This leads to the use of a number of relatively common terms, which are irrelevant for characterizing the scope of a conference. To eliminate such unwanted terms, we use the term strength method from [12]. This method is based on the idea that terms shared by closely related documents are more informative than others. The strength of a term w is thus computed by estimating the probability that a term w occurs in a document d_1 given that it occurs in a related document d_2:

$$strength(w) = P(w \in d_1 | w \in d_2) \tag{6}$$

If there is no information available regarding the relatedness of the documents, as in our case, the pairs of related documents (d_i, d_j) must first be identified. To this end, we have simply used method *tfidf-txt* from Section 3.1, and the pairs with a similarity degree above a certain value v are considered as related. We calculate v iteratively using a threshold γ, which represents the average number of documents we want each document to be related to (i.e., the average number of pairs (d_i, d_j) for each d_i). First, a random value between 0 and 1 is set as initial value of v, and all documents are compared. If the average number of related documents per document is above γ (i.e., each document is related to too many documents), the value of v is raised, and the process is repeated until the average number of related documents is below γ. Since a too small number of related documents is not desirable either, a second threshold γ' can be used to prevent that. In our case we set $\gamma = 20$, $\gamma' = 10$ as satisfactory performance is achieved in that range [12].

After calculating $strength(w)$ for every term w in the CFP collection, the N strongest terms are selected, ignoring the rest. For our experiments in Section 4 we have used $N = 500$ and $\mathcal{C}_{txt}$, since that combination performed well in early tests. The documents are then modelled as in Sections 3.1 and 3.2. When referring to particular methods in Section 4.2 below, we indicate when feature selection was used by adding the suffix *-fs* to the name of the method.

3.4 Related Authors

To reflect users' interest for those conferences whose PC members they are familiar with we propose to calculate extra models exclusively based on papers and compare them. Specifically, we compare the CFP model based on the concatenation of the abstracts of the papers written by the PC members (d_{con}) with a user model based on the concatenation of the abstracts of the papers written by the researchers usually cited by that particular user (d_{aut}). Depending on the type of model, the CFP model based on d_{con} is made according to method *tfidf-con* or *lm-con*. For the user model based on d_{aut} we simply replace d_{con} by d_{aut}^{con} in the definitions of those methods.

The method used to create and compare these extra models is always analogous to that used to calculate the original result, e.g. if the original result is obtained with method *lm-txt*, method *lm-con* is used to calculate these extra models, and they are then compared using the Kullback-Leibler divergence.

The idea is to use these models to complement the result obtained with the methods seen in the previous sections. In particular, once the models are created and compared, we simply combine the result with that of the original comparison by means of the weighted average. For example, to compare CFP *cfp* and user u with method *tfidf-txt*, the result was given by $sim_c(\mathbf{cfp_{txt}}, \mathbf{u_{txt}})$. However, if we take into account these extra models based on d_{con} and d_{aut} (in this case, cfp_{con} and u_{aut}), the result is now given by:

$$\alpha \cdot sim_c(\mathbf{cfp_{txt}}, \mathbf{u_{txt}}) + \beta \cdot sim_c(\mathbf{cfp_{con}}, \mathbf{u_{aut}}) \tag{7}$$

where $\alpha + \beta = 1$. Based on preliminary experiments, we use $\alpha = 0.8$ and $\beta = 0.2$ for the experiments in Section 4.2. We indicate that these extra models are used by adding the suffix *-nam* to the name of the method.

3.5 Related Authors and Feature Selection

Finally, both previously introduced variations can be combined: first, feature selection is applied, which also reduces the number of terms in the extra models based on the frequently cited authors, and then, as explained in the previous subsection, the models are compared separately, to finally combine the results. We indicate that this variation is used by adding the suffix *-fn* to the name of the method.

4 Experimental Evaluation

4.1 Experimental Set-Up

To build a test collection and evaluate the proposed methods, we downloaded 1769 CFPs from DBWorld, which reduced to 1152 CFPs after removing duplicates. Additionally, those CFPs lacking an introductory text or an indicative list of topics were removed too, which further reduced the total number to 969 CFPs. Each of these CFPs has a text part (union of introductory text and topics) and a concatenation of the abstracts of the papers written by the PC members in the last 2 years[6], where available.

On the other hand, 13 researchers from a field which relates to the scope of DBWorld took part in our experiments as users. In order to profile them, we downloaded the abstracts of the papers they wrote in the last 5 years. The ground truth for our experiments is based on annotations made by these 13 users. In a first experiment, each user indicated, for a minimum of 100 CFPs, whether these were relevant or not (relevance degree of 1 or 0 respectively). Then, using each of the studied methods, the CFPs annotated by the users were ranked such that ideally the relevant CFPs appear at the top of the ranking.

In a second experiment, we considered only CFPs assessed as highly relevant by at least one of the methods. To this end, we selected for each user and each of the 24 studied methods the top-5 CFPs of the rankings obtained in the first experiment. This resulted in 120 CFPs, which reduced to an average of about 50 CFPs per user due to overlap between the top-5 CFPs returned by each method. Each of those CFPs was then rated by the user, who gave them a score between 0 ("totally irrelevant") and 4 ("totally relevant"). Again, using each of the studied methods, these CFPs were ranked such that ideally the most relevant CFPs appear at the top of the ranking.

To evaluate the rankings resulting from both experiments, for each user and each method we use normalized discounted cumulative gain (nDCG) [3] to measure the relevance of each CFP according to its position in the ranking. The idea

[6] All the information regarding research papers was retrieved from the ISI Web of Science, `http://apps.isiknowledge.com`

of this measure is that the greater the ranked position of a relevant document, the less valuable it is for the user, as users tend to examine only those documents ranked high, except if those documents do not satisfy their information needs, in which case it is more likely that they still consider lower ranked documents. This is reflected by the discounted cumulative gain of the document ranked in position r:

$$DCG_r = rel_1 + \sum_{i=2}^{r} \frac{rel_i}{log_2 i} \tag{8}$$

The relevance rel_i of the document ranked in position i is the relevance indicated by the user, i.e. 0 or 1 for the first experiment, and 0, 1, 2, 3 or 4 for the second experiment.

Since the number of CFPs annotated by each user might be different, the length of the obtained rankings varies. In order to compare the DCG values we need to calculate the normalized DCG:

$$nDCG_r = \frac{DCG_r}{iDCG_r} \tag{9}$$

where $iDCG_r$ is the ideal DCG at position r: the DCG obtained at position r in the ideal case where all documents are perfectly ranked, from most to least relevant, according to the users' annotations.

For both experiments in Section 4.2 we work with the nDCG of the CFP ranked in the last position, i.e. nDCG_r where r is the total number of CFPs in the ranking, as this value reflects the gains of all the CFPs throughout the whole ranking.

4.2 Results

Tables 1 and 2 summarize the results of the first and second experiment respectively. In particular, for each method we show the average nDCG_r for the 13 users, where r is the number of CFPs in the ranking for each user as indicated in the previous section. For the sake of simplicity we have used some fixed values for the λ parameters of (5) in the methods based on language modeling. In particular, we use $\lambda_1 = 0.9$ and $\lambda_2 = 0$ for the *lm-txt* method (i.e. analogously to *tfidf-txt*, it only uses the information from the text parts of the CFPs); $\lambda_1 = 0$ and $\lambda_2 = 0.9$ for the *lm-con* method; and $\lambda_1 = 0.4$ and $\lambda_2 = 0.4$ for the *lm-tot* method.

First we compare the different kinds of information that can be used: introductory text plus topics, concatenation of the abstracts of the papers recently written by the PC members, or the concatenation of both. Figures 1 and 2 show that using the abstracts alone (*con*) does not suffice to outperform the methods based on the textual content (*txt*), while those based on the concatenation of abstracts and textual content (*tot*) seem to perform comparably or slightly better than the *txt* methods, except for the language model based methods without feature selection. If we fix the method and the use of feature selection or abstracts written by frequently cited authors, we can see that the differences,

Table 1. Ranking of methods for the first experiment, nDCG values

Method	nDCG	Method	nDCG	Method	nDCG
tfidf-tot-fs	0.606	tfidf-con-fn	0.553	lm-txt	0.51
tfidf-tot-fn	0.599	tfidf-con-nam	0.551	lm-tot	0.493
lm-tot-fs	0.575	tfidf-con-fs	0.549	lm-con-fs	0.493
tfidf-tot	0.563	tfidf-con	0.544	lm-txt-nam	0.482
tfidf-txt-fn	0.563	tfidf-txt	0.542	lm-con-fn	0.469
tfidf-txt-nam	0.562	lm-txt-fs	0.529	lm-con	0.44
tfidf-tot-nam	0.561	lm-tot-fn	0.516	lm-tot-nam	0.436
tfidf-txt-fs	0.555	lm-txt-fn	0.512	lm-con-nam	0.421

Table 2. Ranking of methods for the second experiment, nDCG values

Method	nDCG	Method	nDCG	Method	nDCG
lm-tot-fs	0.745	lm-txt	0.691	tfidf-con-fn	0.646
tfidf-txt-nam	0.728	lm-tot-fn	0.686	tfidf-con	0.637
tfidf-txt	0.715	lm-txt-fn	0.682	tfidf-con-nam	0.636
tfidf-tot-fs	0.713	tfidf-tot	0.661	lm-con-fs	0.606
tfidf-txt-fn	0.707	lm-tot	0.653	lm-con-fn	0.587
tfidf-tot-fn	0.706	tfidf-con-fs	0.649	lm-tot-nam	0.566
tfidf-txt-fs	0.705	tfidf-tot-nam	0.648	lm-con	0.555
lm-txt-fs	0.700	lm-txt-nam	0.647	lm-con-nam	0.502

although real, are not significant[7] enough, except for *tfidf-con-nam*, *lm-con* and *lm-con-fs*, which perform worse than *tfidf-tot-nam*, *lm-tot* and *lm-tot-fs* in Experiment 1. In Experiment 2, the differences among all language model cases are significant except for those between *lm-txt-fs* and *lm-con-fs*, *lm-txt-fn*/*lm-con-fn*, *lm-txt*/*lm-tot*, and *lm-txt-fn*/*lm-tot-fn*. On the contrary, the differences in the vector space model cases are not significant except for *tfidf-tot*/*tfidf-con*, and those between the methods involving *nam*.

To study the impact of feature selection (*fs*), the models based on frequently cited authors (*nam*) and the combination of both (*fn*), we fix the method and the type of information used. In general, the best results are obtained when feature selection is applied. It must be noted, however, that these differences are only significant in some cases: the *con* and *tot* cases of the language model based methods in Experiment 1, and the *tot* case of the language model based methods in Experiment 2. On the other hand, results obtained with the *nam* methods are worse than the original, with significant differences for the language model based methods in Experiment 2. As for *fn*, it usually improves the original results, but there is no significant evidence of this.

Finally, we compare the methods based on the vector space model with those based on language modeling. In Figures 1 and 2 we can observe that the former generally outperform the latter. Some methods based on language modeling (*lm-txt-fs* and *lm-tot-fs* in Experiment 1, joined by *lm-txt*, *lm-tot*, *lm-txt-fn* and

[7] In this work we consider a difference to be significant when $p < 0.05$ for the Mann-Whitney U test.

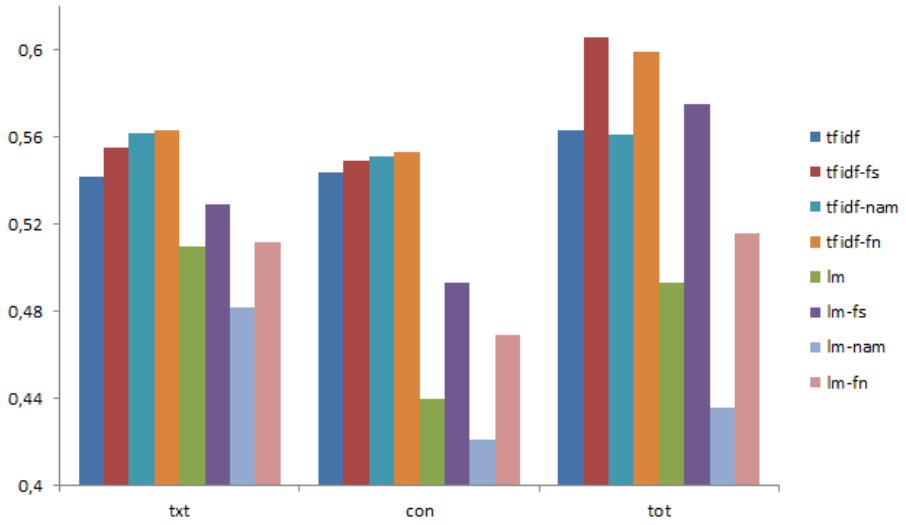


Fig. 1. Results for experiment 1; the Y-axis shows the nDCG, while the X-axis indicates the kind of information used

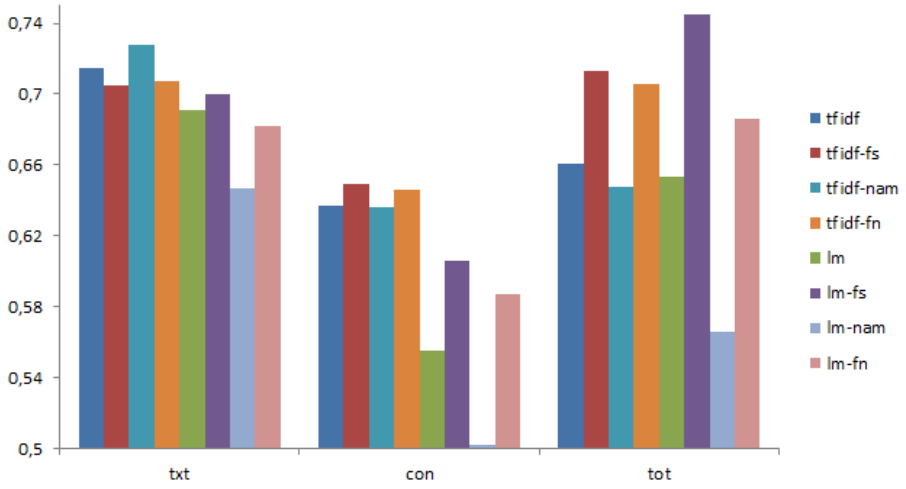


Fig. 2. Results for experiment 2; the Y-axis shows the nDCG, while the X-axis indicates the kind of information used

lm-tot-fn in Experiment 2) perform comparably to those based on the vector space model, but although both vector space model and language model based approaches can achieve good results, the former appear to be much more robust against changes in the particular way in which CFPs are modelled. In a comparison where the information type and the use of feature selection is fixed (e.g. we compare *tfidf-txt-fs* and *lm-txt-fs*) methods based on the vector space model significantly outperform those based on language modeling in many cases. In Experiment 1 this is the case of *txt-nam*, *txt-fn*, the *tot* methods except *tot-fs*, and all *con* methods. On the other hand, in Experiment 2 differences are significant for *txt-nam*, *tot-nam*, and the *con* methods except *con-fs*.

5 Conclusion

We have proposed and compared several content-based methods to match users with CFPs. We have studied the impact of the different types of information available, the accuracy of the models that represent such information, and the effect of feature selection on these models. Also, using the users' names and the names of the CFP members we have accessed the papers recently written by them to profile the users and to complete available information about the CFP respectively. Information about authors frequently cited by the users is also used to reflect the importance given by the users to the CFPs of conferences with people in the PC working in the same field and whose work they usually cite.

The results indicate that methods based on the vector space model are generally more robust, and achieve the best performance on this task. Both for vector space models and language models, feature selection improved the results.

Finally, we have also seen that although the abstracts of the papers written by the PC members can be helpful when combined with other information, the resulting models are not sufficiently accurate to be used on their own.

As mentioned in the introduction, we remark that content-based approaches alone do not suffice to cover all the aspects of the task of matching users and CFPs, as the relevance of a conference depends also on information not contained in the text of the CFPs. Therefore, the studied content-based methods should be complemented with other techniques, which provides an interesting starting point for future work. Collaborative filtering would be of great help; in this case a given CFP could be matched to a user because another user with similar interests attended a previous edition of that conference. Alternatively, a user could get a notification about a CFP because that given conference covers similar topics as another conference he attended in the past. Also, trust-based methods could reflect additional information not covered by collaborative filtering: a user could then be notified about a conference because a researcher he trusts is in the program committee, or because he trusts the conference given its impact on his research field.

References

1. Bogers, T., van den Bosch, A.: Recommending scientific articles using CiteULike. In: Proc. of the 2008 ACM Conf. on Recommender Systems, pp. 287–290 (2008)
2. Deng, H., King, I., Lyu, M.R.: Formal models for expert finding on DBLP bibliography data. In: Proc. of the 8th IEEE International Conference on Data Mining, pp. 163–172 (2008)
3. Järvelin, K., Kekäläinen, J.: IR evaluation methods for retrieving highly relevant documents. In: Proceedings of the 23rd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 41–48 (2000)
4. Karimzadehgan, M., Zhai, C., Belford, G.: Multi-aspect expertise matching for review assignment In. In: Proc. of the 17th ACM Conference on Information and Knowledge Management, pp. 1113–1122 (2008)
5. McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the recommending of citations for research papers. In: Proc. of the 2002 ACM Conf. on Computer Supported Cooperative Work, pp. 116–125 (2002)
6. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
7. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proc. of the 21st Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 275–281 (1998)
8. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24, 513–523 (1988)
9. Strohman, T., Croft, W.B., Jensen, D.: Recommending citations for academic papers. In: Proc. of the 30th Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 705–706 (2007)
10. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence 2009, 4:2–4:2 (2009)
11. Tang, W., Tang, J., Tan, C.: Expertise Matching via Constraint-Based Optimization. In: Proc. of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 1, pp. 34–41 (2010)
12. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proc. of the 14th International Conference on Machine Learning, pp. 412–420 (1997)

Domain Adaptation of General Natural Language Processing Tools for a Patent Claim Visualization System

Linda Andersson, Mihai Lupu, and Allan Hanbury

{andersson,lupu,hanbury}@ifs.tuwien.ac.at

Abstract. In this study we present a first step towards domain adaptation of Natural Language Processing (NLP) tools, which we use in a pipeline for a system to create a dependency claim graph (DCG). Our system takes advantage of patterns occurring in the patent domain notably of the characteristic of patent claims of containing technical terminology combined with legal rhetorical structure. Such patterns make the sentences generally difficult to understand for people, but can be leveraged by our system to assist the cognitive process of understanding the innovation described in the claim. We present this set of patterns, together with an extensive evaluation showing that the results are, even for this relatively difficult genre, at least 90% correct, as identified by both expert and non-expert users. The assessment of each generated DCG is based upon completeness, connection and a set of pre-defined relations.

Keywords: Graph visualization, domain adaptation, Natural Language Processing.

1 Introduction

The overall aim of this novel Dependency Claim Graphs (DCG) system is to support the cognitive process of reading and interpreting the claim text of a patent document. A patent document consists of four main textual components (title, abstract, description, and claim), intended to fulfil different communication goals. The claim has its own very special conceptual, syntactic and stylistic/rhetorical structure. It needs to be composed in such a way as to completely describe the essential component of the invention, while making patent infringement difficult [1].

In the IR community, the research focus has mainly been on improving and developing methods and systems for supporting patent experts in the process of Prior Art search (i.e. retrieving patent documents which could invalidate the patent) [2]. There have been two main evaluation campaigns (NTCIR and CLEF-IP) with patent-related tasks, while others (e.g. TREC-CHEM) have used substantial patent collections [3]. Less research attention has been given to other information processing activities conducted by the professional patent searchers. In order to formulate complex search queries, the patent experts extract phrases and terminologies used in the patent application. Part of this pre-search analysis consists of examining the claim section, to define scope and limitation [2, 4].

Additionally, during a post process analysis, patent claims are important in order to establish similarity between different patents. This motivates our efforts in developing

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 70–82, 2013.

a system that supports the information analysis process by visualizing differences and similarities within a patent claim in order to show different aspects of the invention.

In order to generate claim graphs for the entire claim as well as for each paragraph, we use several different layers of linguistic information: Part-of-speech (PoS-tagger), phrase boundaries (chunker) and discourse theory. Instead of using a full-scale parser assigning syntactic relation, we use generic lexico-syntactic patterns for Relation Extraction (RE). We also use lexico-syntactic patterns to adapt the analysis from the NLP tools used in the pipeline to better reflect the syntax of claims sentence.

There are two main reasons for using lexico-syntactic patterns instead of a full-scale parser: first, the lack of robustness of the parser tagger and chunker, and second, the speed. Moreover, if we just change the focus from the mainstream genre such as newspaper articles to more specific corpora, several of the existing tools show a significant decrease in performance [5]. In a German study the accuracy decreased approximately by 5 percentage units (97% to 92%) when training on ideal German corpora and then testing on German Web corpora [6].

There are few NLP tools adapted for the patent domain. Furthermore, such tools are generally restricted to only working on pre-defined technical fields, as in [7] or [8]. Since the aim with the DCG system is to support the cognitive process of reading claims, the NLP applications used in the pipeline are required to handle all types of claims, as well as all technical fields. Therefore we investigate the use of generic lexico-syntactic patterns.

We used the English part of the CLEF–IP 2012 Passage Retrieval topic set as test collection, since the technical field distribution reflects the collection composition.

The rest of the paper is organized as follows. Section 2 gives insight to the related work and linguistic characteristics of the patent genre. Our method, the experiment and evaluation schema are presented in Section 3. In Section 4 the outcome of the evaluation task is presented along with analysis of general errors made by the NLP tools used in the pipeline. Conclusion and final remarks are given in Section 5.

2 Related Work

In order to create support tools for the patent expert we need to understand their daily work task, as well as the patent lifecycle and the linguistic character of this text genre [4].

The patent claims define the technical boundaries that should be protected by the patent. Therefore, the claims vocabulary consists of terms with legal impact as well as technical terms. The rhetorical structure of a claim is pre-defined into three parts: preamble, transitional phrase, and body. Ferraro [9] gives a more linguistic description of each part's function compared to the patent regulation literature. Here, we choose to use Ferraro's definition. The preamble is the claims introduction clause, which could include the main function of the invention as well as its purpose and field. The transitional phrase (or linking words), connect the preamble to the part specifying the invention itself (the body). In the transitional phrase words such "comprising", "containing", "including", "consisting of", "wherein" and "characterized in

that" are frequently used. The body explains the invention and enumerates the legal and technical limitations.

Claims can be divided in two different categories: independent and dependent claims. An independent claim is a legal statement of its own, and does not refer back explicitly or implicitly to any other claims. A dependent claim depends on the claim/claims it explicitly refers back to by phrases such as "according to claim 1", "according to any of the previous claims", etc. Figure 1 displays the entire claim section and a claim tree according to the European Patent Office (EPO) existing tree claim structure of the patent application EP1306390 (A1).

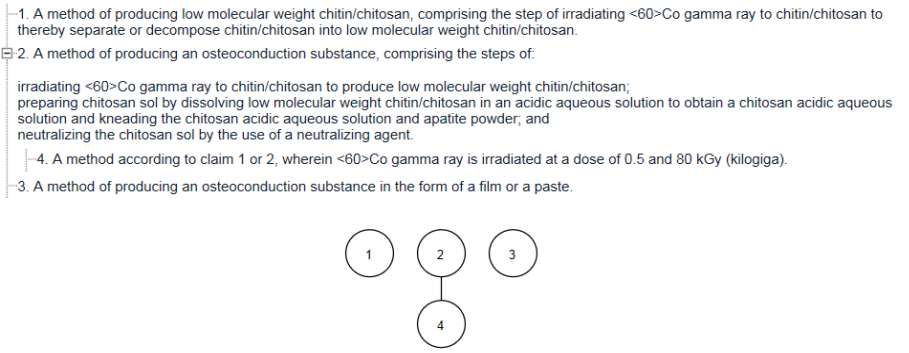


Fig. 1. Claim tree of Patent Application EP-1306390 (A1)[1]

The application EP1306390 (A1) consists of three independent claims (1, 2, 3) and one dependent claim (4), as visualized in Figure 1. In comparison with the EPO claim tree which make use of explicit reference in the text to other claims i.e. "according to previous claims", "according to claim 2 and 1" etc. our DCG system takes advantage of implicit reference by detecting discourse references (e.g "the chitosan acidic aqueous solution" is referring to the same entity as "a chitosan acidic aqueous solution"). However, in order to detect noun phrases (NP) we first need to parse each sentence in order to identify given entities. Figure 2 shows the sentence claim graph of claim 1.

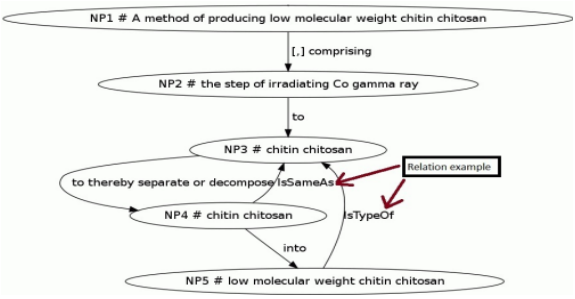


Fig. 2. Claim 1 of Patent Application EP-1306390 (A1)

[1] http://bit.ly/19z9t7V

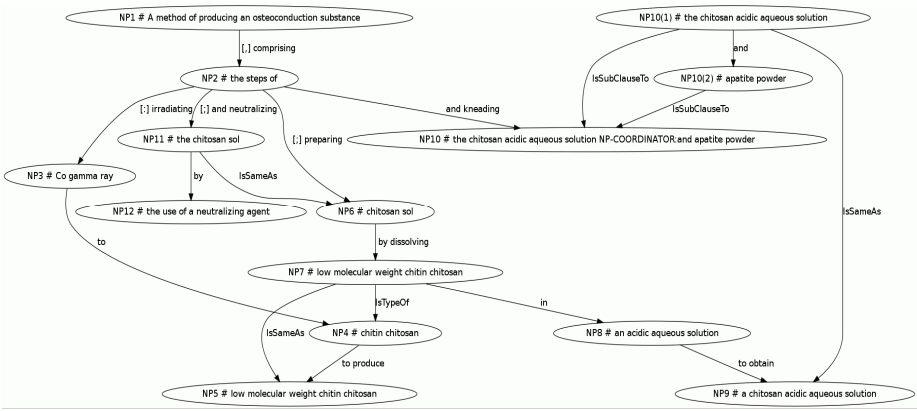


Fig. 3. Claim 2 of Patent Application EP-1306390 (A1)

The aim of the graph for each claim sentence is to identify the boundary of the main NP, as well as assign discourse relations such as IsSameAs and IsTypeOf. Each node is assigned a sequence marker NP1, NP2 etc. The more complex claim sentence (i.e. Claim 2) is presented in Figure 3.

In Figure 3 we can also see a larger NP constituent, consisting of a set of smaller NPs combined with a coordinator and collapsed into node NP10. The smaller NPs (Sub nodes) are linked by an IsSubClauseTo relation to the main node. The linear order of the sentence is changed since the sentence has a transitional phrase that indicates a listing of activities.

In the full scale DCG system, each claim sentence tree will be interactively connected to other claims when the user requests links based upon different pre-defined relations such as explicit dependency relations (i.e. IsDependencyOf), IsSameAs and IsTypeOf. However, in this present paper we evaluate the effectiveness of the preprocess i.e. the construction of the tree per claim sentence.

2.1 Characteristics of the Patent Claim Text

Verberne et al [10] presented a comparative linguistic genre study comparing patent claims text and sentences with general language resources. They found that the sentences of patent claims (allowing semicolon and colon as sentence splitter) were generally longer than the sentences found in the British National Corpus (BNC). The patent sentences had a median of 22 tokens and an average length of 53 tokens (based upon a comparative study consisting of 581k sentences). Ferraro [7] reported it is not unusual to have sentences in the claim section consisting of 250 tokens; and in Wäschle and Riezler [11] it was also reported that a corpus of 500k claim sentences consisted of approximately 18,355,584 tokens and 270,013 types and the average type-token ratio (TTR) is 0.0147. The TTR indicates the variation in vocabulary distribution, which needs to be handled in the NLP pipeline.

A few domain adaptations of NLP tools have relied on incorporating domain knowledge in the NLP process by extracting terms from patent collections [1, 12]. However, to just increase the lexical coverage will not solve the problem, since token coverage is only part of the problem. Verberne et al [10] concluded there were no significant differences between general English and the English used in patent claim text for single token coverage, the (new) technical terminology is more likely introduced on the multi-word level consisting of complex NPs. Also, the literature addressing terminology extraction confirms that the majority of the technical dictionaries consist of terms with more than one word [13]. The technical multi-word phrases consist of noun phrases containing common adjectives, nouns and occasionally prepositions (e.g. 'of'). Therefore, it is important that the focus for domain adaptation lies within identifying correct noun phrase boundaries as well as increasing of lexical coverage.

2.2 Domain Adaptation and Evaluation

In [1], the aim was to reduce complexity in claims and increase readability. It required modification of the pre-processing, training of a super tagger (lexical driven) and additional domain rules to define dependency relations. Previous work addressing claim readability has been conducted on Japanese patent claims [14].

The retrieval system PHASAR has been domain adapted towards the patent domain by increase of lexicon coverage [12]. The system integrates linguistic notation in the search mechanism, it uses linguistic information and displays linguistic knowledge to the searcher. The system aims to capture dependency relationships between words via dependency triples. PHASAR uses a special grammar based on AEGIR (an extension of Context Free grammar formalism) adapted for robust parsing to be used in IR.

In terms of evaluation, to the best of our knowledge, only small-scale linguistic evaluation of parsers has been conducted. In [10], a comparison between two different parsers was performed: AEGIR used in the PHASER system, and the Connexor CFG parser. Conducting a linguistic evaluation of the performance of a parser or part of speech application is a time consuming task and requires both linguistic expertise as well as domain knowledge. In [10], 100 randomly selected short patent sentences (5-9 words) were assessed based upon generated dependency triples; the F1-scores for AEGIR 0.47 and for Connexor CFG 0.71 were calculated. The inter-annotator agreement was 0.83 and was computed by counting the number of identical triples divided by the total number of triples created by another annotator.

In another study, Parapatics and Dittenbach also aimed to reduce the complexity in claims [8], the General Architecture for Text Engineering (GATE) was used to decompose sentences by identifying the claim-subject and assigning dependent claims to the correct independent claim. The Stanford dependency parser was used in the large evaluation (5000 claim sentences), but only its performance in terms of ability to parse and its memory usage was assessed for the decomposed and for the original sentences. The correctness of the parsed sentences was never investigated.

In Ferraro [7], Minipar was used as part of the NLP-pipeline to extract verbal relations in patent claims. No evaluation of the parser performance on the patent text was conducted, only citing the performance of Minipar on the Susanne Corpus[2] (0.89 precision). However, before the final parsing of the patent sentence, the sentence was decomposed to smaller units by using a domain adapted segmentation tool, as well as a rules driven algorithm for paraphrasing the segment into complete sentences [15].

3 Our Approach

The English part of the CLEF–IP 2012 Passage Retrieval topic set was used as training and test set. For training purposes we randomly selected 40 claim sentences, which we manually investigated. The test set consisted of 600 randomly selected claim sentences. In Table 1 the average number of tokens and types, as well as the average TTR is shown for the test set, divided per International Patent Classification system (IPC) section. In parenthesis, for each average, we indicate the standard deviation, to give an idea of the range in the entire population. We follow the same convention throughout this paper.

Table 1. Token, type TTR distribution for Test Set

IPC	Token Average (stddev)	Type Average (stddev)	TTR Average (stddev)
A (Human Necessities)	33.37 (17.10)	26.34 (9.55)	0.84 (0.13)
B (Performing Operation; transporting)	33.59 (20.96)	26.37 (11.75)	0.85 (0.13)
C (Chemistry; Metallurgy)	28.60 (14.91)	23.95 (9.36)	0.88 (0.11)
D (Textiles; Paper)	41.53 (20.29)	31.81 (11.76)	0.81 (0.11)
F (Mechanical Engineering; Lighting; Heating,Weapons; Blasting)	38.75 (25.00)	29.29 (13.62)	0.81 (0.12)
G (Physics)	30.26 (19.25)	23.99 (10.58)	0.86 (0.13)
H (Electricity)	38.16 (23.33)	27.89 (12.21)	0.81 (0.85)
Total	33.21 (19.31)	26.16 (10.88)	0.85 (0.13)

In a previous work addressing phrase retrieval [16], an observation study of noun phrase patterns was conducted in order to define noun phrase boundaries in the patent domain.

However, the DCG system requires more flexibility in the phrase boundary than given by the lexico-syntactic pattern. Therefore we chose to use the baseNP Chunker [17]; and construct generic rules modifying the output of the chunker based upon previously observed patterns. All sentences were annotated with the Stanford Part-of-Speech tagger, using the english-left3words-distsim.tagger model [18].

[2] http://www.grsampson.net/SueDoc.html

Table 2. Assessment of evaluation parameters by all assessors

Rule	Original NP Sequence	Modified NP Sequence	Modifying
"said" as an article	said/VBD [supercritical/JJ fluid/NN]	[said/VBD supercritical/JJ fluid/NN].	PoS-tagger
preposition within the preamble phrase	[The/DT soccer/NN shoe/NN] of/IN [claim/NN 4/CD]	[The/DT soccer/NN shoe/NN of/IN claim/NN 4/CD]	Chunker
include present participle	[A/DT method/NN] of/IN fabricating/VBG [a/DT semi-conductor/NN device/NN]	[A/DT method/NN of/IN fabricating/VBG a/DT semi-conductor/NN device/NN]	Chunker
infinitive verb tagged as NN	[said/VBD laser/NN radia-tion/NN] to/TO [exit/NN] [said/VBD exit/NN system/NN]	[said/VBD laser/NN radia-tion/NN] to/TO exit/VB [said/VBD exit/NN sys-tem/NN].	PoS-tagger
include digits into the NP	NP [The/DT method/NN of/IN any/DT of/IN claims/NNS] [12/CD to/TO 16/CD]	[The/DT method/NN of/IN any/DT of/IN claims/NNS 12/CD to/TO 16/CD]	PoS-tagger
list of NPs	in [the/DT group/NN] consist-ing/VBG of/IN [a/DT photore-sist/NN] ,/, [a/DT photore-sist/NN residue/NN] ,/, and/CC [a/DT combination/NN]	into [the/DT group/NN] consisting/VBG of/IN [a/DT photoresist/NN ,/, a/DT photoresist/NN resi-due/NN ,/, and/CC a/DT combination/NN]	Claims dis-course adap-tation spe-cific rules
	A sub rule to 7, Identifying, transition phrases listing sub clauses as seen in figure 2		

We created 9 main rules to adjust and adopt the output from the PoS-tagger and chunker to better reflect the patent domain (see Table 2).

These rules were implemented in order to generate a connected graph and to better identify the NP boundaries occurring in patent claims. Before submitting the claims text to the PoS-tagger and baseNP Chunker, a generic abbreviation handler and a modified sentence splitter were applied. Also, special signs were removed.

3.1 Nodes and Relations

The graph nodes reflect the sentence noun phrases, where we have chosen to collapse NPs consisting of smaller NPs into larger complex NPs, as seen in Figures 2 and 3. The relations (or links) consist mostly of the sentence's verb, preposition or other transition function such as clause markers (',', ";","":"). We also identify three inferred relations: IsSubClauseTo, IsSameAs and IsTypeOf.

If a noun phrase consists of several sub NPs combined with ',' or/and a coordinator, the noun phrase is kept joined, and all sub clauses are also given an inferred

relation IsSubClauseTo to the main NP (see Figure 3). Each sub clause is also given the sequence number of the main NP and a sub sequence number in the order they appear in the main NP e.g. NP10(1). The objective with inferred relation IsSubClauseTo is to visualize list of NPs.

IsSameAs and IsTypeOf are partly associated with the discourse structure of the claims sentence. For IsSameAs, identifying the same entity, only the initial article/word may differ between 'a', 'said', or 'the' (see Figure 2). The relation IsTypeOf is assigned when one node has been pre-modified but the head noun of both NPs is the same as in Figure 2 where NP5 "low molecular weight chitin chitosan" IsTypeOf to NP3 "chitin chitosan". The nodes and links are made into RDF triples, subsequently used in order to generate the sentence claims graphs[3].

3.2 Evaluation

To the best of our knowledge, there is no existing gold standard for any type of linguistic annotated information in the patent domain. Therefore, the assessment was based on manually assessing all graphs. Due to the fact that there are very few people having the level of deep linguistic knowledge, as well as the domain specific knowledge required to conduct assessment of linguistic accuracy of the displayed graphs, we decided upon a more generic evaluation schema.

The assessor group was divided into two groups: expert (3) and non-expert (14). The expert group consisted of linguists with some existing knowledge of the patent domain. The group of non-experts consisted of engineers and university students.

For the evaluation task, we constructed a simple interface showing the graph as well as the original sentence (see Figure 4).

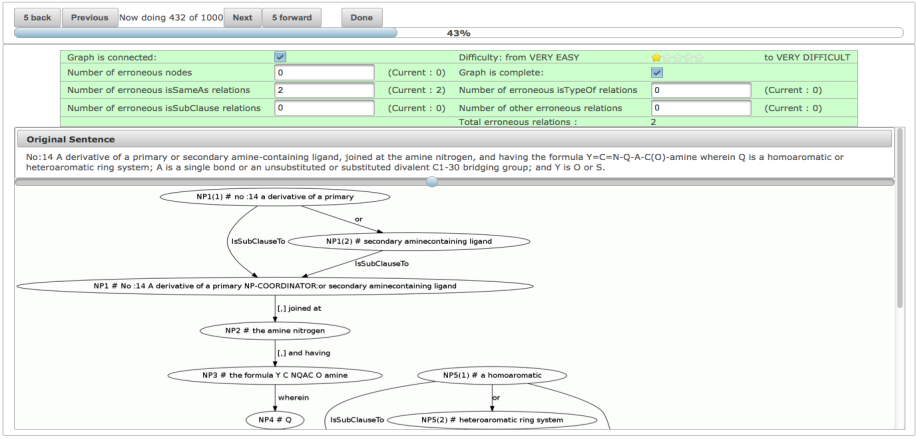


Fig. 4. Evaluation Tool

[3] Software: Graphviz (`http://search.cpan.org/~rsavage/GraphViz-2.14/lib/GraphViz.pm`)

Each claim graph was randomly given to one expert and one non-expert. Since the task turned out to be very time consuming, not all of the assessors assessed all of the claim graphs. All non-experts were assigned 100 claim graphs.

The assessment task consists of:

i) assessing the completeness of each displayed sentence claim graph where each word in the original sentence should also be represented in the graph.
ii) assessing connection, i.e. the graph must consist of a single connected component (ignoring edge directions).
iii) assessing if the claim graph displays correctly identified nodes and relations.

We defined seven parameters we asked the assessors to assess for each graph:

1. graph is complete,
2. graph is connected,
3. number of erroneous nodes,
4. number of erroneous IsSameAs relations,
5. number of erroneous IsSubClauseTo relation,
6. number of erroneous IsTypeOf relations,
7. number of other erroneous relations.

In the instructions for the evaluation task, a simple example and a domain example were given for each type of relation and erroneous nodes and relations. In order to find out how difficult assessors found the tasks to be, we asked each assessor to grade each graph from as scale 1 (very easy) to 5 (very difficult).

4 Results

We computed the inter-annotation agreement between expert and non-expert users for each assessment parameter. Similarly to [10], we also computed the inter-annotator agreement as the percentage of equal assessments among all pairs of assessments (Table 3).

Table 3. Inter-annotation agreement

Assessor Pair	No of sentences	Connected Graphs	Erroneous Nodes	Erroneous IsSameAs	Erroneous IsTypeOf	Erroneous IsSubClauseTo	Erroneous Other Relations	Complete graphs	Graph Difficulty
Non-expert vs Expert	182	98.35	68.13	87.91	97.80	96.15	69.78	84.62	26.37
Expert vs Expert	193	97.41	61.14	84.97	97.93	98.45	64.77	74.09	56.48

As seen in Table 3, the inter-annotation agreement of grading how difficult the task was for each graph was low between experts and non-experts (26.37%), while the experts agreed 56.48% on the difficulty associated with the task. The disagreement on difficulty does not come as a surprise, due to the wide range of options (1-5) as well as due to the graphs' complex structures, some of them containing up to 26 nodes and relations. Linguists are more trained to this type of visualization of sentences, which makes the interpretation and the reading of the graph more straightforward.

Assessing if the graph was connected, i.e. if for any pair of nodes in the graph there exists a path between them (potentially ignoring edge directions), was shown to be the easiest task when comparing inter-annotation agreement between pairs of non-expert vs expert users (98.35%) and expert vs expert (97.41%) users. For the other parameter assessments, the inter-annotation agreement between the pairs differs approximately by 10 percentage units. For the Erroneous Nodes and Erroneous Other Relations, there is a considerable drop in the agreement between both pairs as seen in Table 3. Compared to the other parameters assessed, these are by far the most loosely defined for the non-expert user. Furthermore, these two are strongly correlated – it is often the case that when one node is erroneously identified, an erroneous relation (other than IsSameAs, IsTypeOf, or IsSubClauseTo) is added to the graph.

As seen in Table 4 out of the 600[4] graphs evaluated most of the graphs are connected (for all assessors 97%, for only expert assessors 97%) and complete (for all assessors 86%, for only expert assessors 81%). Among the three inferred relations, IsSubClauseTo and IsTypeOf had the fewest errors. The algorithm extracting the IsSameAs relation was mistakenly identifying split chemicals as same units as well as identifying a longer sequence node with a smaller node containing a sub-sequence of the node.

Table 4. Assessment of evaluation parameters averaged over all assessors

IPC	# Sentences	Erroneous Nodes	Erroneous IsSameAs	Erroneous IsTypeOf	Erroneous IsSubClauseTo	Erroneous Other Relations	Complete Graph	Connected Graph	Difficulty
A	291	0.05 (0.10)	0.03 (0.07)	0 (0.02)	0.01 (0.04)	0.03 (0.09)	0.87 (0.31)	0.98 (0.13)	2.08 (1.16)
B	277	0.08 (0.11)	0.02 (0.06)	0 (0.01)	0 (0.01)	0.06 (0.10)	0.86 (0.32)	0.98 (0.14)	2.03 (1.14)
C	284	0.07 (0.10)	0.02 (0.07)	0 (0.01)	0 (0.03)	0.05 (0.09)	0.86 (0.32)	0.97 (0.16)	1.99 (1.13)
D	52	0.04 (0.08)	0 (0.02)	0 (0.01)	0 (0.02)	0.02 (0.06)	0.93 (0.24)	1 (0)	1.86 (1.04)
F	43	0.09 (0.09)	0.02 (0.05)	0 (0)	0 (0)	0.05 (0.08)	0.81 (0.33)	0.97 (0.17)	1.93 (0.82)
G	163	0.1 (0.12)	0.03 (0.07)	0 (0.01)	0.01 (0.03)	0.07 (0.10)	0.85 (0.34)	0.96 (0.18)	2.04 (1.16)
H	102	0.09 (0.10)	0.03 (0.06)	0.01 (0.02)	0 (0.01)	0.06 (0.09)	0.79 (0.39)	0.93 (0.25)	2.38 (1.31)
Total	1212	0.07 (0.11)	0.02 (0.07)	0 (0.02)	0 (0.03)	0.05 (0.09)	0.86 (0.32)	0.97 (0.16)	2.05 (1.15)

The Erroneous Nodes and Erroneous Other Relations is an indication when a noun phrase boundary has been wrongly identified. There were several where part of the relation between NP1 and NP2 "according to claim" contains the word claim that should have been part of the NP2 i.e. [A method] according to [claim 2]. The error

[4] The 1212 figure that appears in the table is due to the fact that many claims are assigned to more than 1 IPC Section.

was caused by the PoS tagger assigning the word "claim" the VB (verb) tag, and therefore affecting the chunker and ultimately the graph. This was triggered by "to", which functions as an infinitive marker. The annotation of the word "claim" by the PoS-tagger in the sequence "according to claim" was unstable, randomly assigning VB (verb) or NN (noun). Only when the word "claim" was written with a Capital letter, the PoS-tagger identifies "Claim" as a Proper noun (NNP) which, given the context, is the most proper PoS-tag.

Furthermore, due to the PoS-tagger assigning words such as the "hydrogenated/VBN" (verb, past participle) instead of JJ (adjective), both erroneous nodes and relations were generated. This error is one of the most common erroneous and unstable annotations of the PoS-tagger, which clearly affects the NP boundary detection. The chunker was unstable in chunking digits into units. For instance the chunker joined the first digits [Claim/NNP 1/CD ,/, 2/CD] but not the last ",/, or/CC [3/CD]". This did not depend on whether a coordinator (CC) was present or not.

Among old patents, there are also some stylistic text representations, which severely damage the text processing tools. For instance, it is common that the word characterized is written with space in between the letter sequence ("c h a r a c t e r i z e d"). This makes the entire graph more or less erroneous. Moreover, the presence of OCR-errors affected the extraction, e.g. the word 'in' mistakenly being identified as the letter 'm'.

5 Conclusion

In this paper, we examined:

i) visualization of patent claim sentences in order to create a system which supports the cognitive process of analyzing patent claims
ii) domain adaptation of the NLP tools used in the pipeline, as well as a generic evaluation schema using non-experts and experts.

Our result shows that approximately 90% of all graphs used in the test collection have been assessed both by expert and non-expert to be complete, connected and having correctly identified nodes and relations. When comparing the inter-annotation agreement for each of the pre-defined erroneous nodes and relations we see that the expert-expert and the expert-non-expert values are similar. Consequently, the inter-annotation agreement indicates that for the pre-defined evaluation scheme, as presented in this study, using non-expert assessors is at least feasible and not as problematic as anticipated, despite the highly specific characteristics of the patent claims.

Our finding confirms that performance decreases when using existing general NLP tools when changing text focus from the mainstream genre text towards a specific text genre. Even if we used a state-of-the-art PoS-tagger for English with high accuracy, even small errors affected the parsing of a patent claims sentence negatively. Despite this, our result indicates that a general PoS-tagger and chunker can be used successfully on patent claims if combined with rules based upon observed syntactic patterns from the patent genre. In order to make the PoS-tagger even more robust when parsing patent text, a more extensive normalization procedure needs to be implemented

dealing with chemical compounds, formulae and OCR-errors, as well as rejoining words written with spaces between their letters.

The general PoS-tagger used in this experiment still made errors, which caused the chunker to generate incorrect phrase boundaries and thereby caused the entire claim graphs to collapse. The average TTR values for each technical field (defined by the IPC section) suggest an alternation of the words distribution for claim sentences. The PoS-tagger can be made more robust by adding more post heuristic rules addressing complex noun phrase constructions. In the future, we will also investigate if post contra rules could be used as an intermediate layer in order to improve the performance of a full-scale parser in the patent domain. The method used to establish the DCG representation could be adjusted to intermediate layers in technical terminology extraction, as well as computing sub graph similarity.

To summarize, in this paper we have presented a DCG construction method applicable to all technical fields of the patent domain. We note that this is particularly important, since a support tool for patent experts needs to be able to deal with variations in terminology and linguistic features. We have also provided the experimental evidence to show that the tool achieves high success rates in identifying the important elements of an invention described in the claims, and the relations that bind them.

References

[1] Sheremetyeva, S.: Natural language analysis of patent claims. In: Proc ACL-2003, Workshop on Patent Corpus Processing, pp. 66–73 (2003)

[2] Hunt, D., Nguyen, L., Rodgers, M.: Patent Searching Tools & Techniques. John Wiley &Sons, New Jersey (2007)

[3] Lupu, M., Huang, J., Zhu, J.: Evaluation of Chemical Information Retrieval Tools. In: Croft, W.B., Lupu, M., Mayer, K., Tait, J., Trippe, J.A. (eds.) Current Challenges in patent Information Retrieval. Springer (2011)

[4] Hansen, P.: Task-based Information Seeking and Retrieval in the Patent Domain: Processes and Relationships. Tampere University Press (Doctoral dissertation), Tampere (2011)

[5] Uematsu, S., Kim, J.-D., Sujii, J.: Bridging the gap between domain-oriented and linguistically-oriented semantics. In: Proc ACL-2009, Workshop BioNLP 2009, pp. 162–170 (2009)

[6] Giesbrecht, E., Evert, S.: Part-of-speech tagging - A solved task? An evaluation of POS taggers for the Web as corpus. In: Alegria, I., Leturia, I., Sharoff, S. (eds.) WAC5 (2009)

[7] Ferraro, G.: Towards deep content extraction from specialized discourse: The case of verbal relation in patent claims Department of Information and communication Technologies: Universitat Pompeu Fabra (Doctoral dissertation) (2012)

[8] Parapatics, P., Dittenbach, M.: Patent Claim Decomposition for Improved Information Extraction. In: Lupu, M., Mayer, K., Tait, J., Trippe, J.A. (eds.) Current Challenges in patent Information Retrieval. Springer (2011)

[9] Ferraro, G., Wanner, L.: Towards the derivation of verbal content relations from patent claims using deep syntactic structures. Knowledge-Based Systems 24(8), 1233–1244 (2011)

[10] Verberne, S., D'hondt, E., Oostdijk, N., Koster, C.: Quantifying the Challenges in Parsing Patent Claims. In: Workshop of AsPIRe, pp. 14–21 (2010)

[11] Wäschle, K., Riezler, S.: Analyzing parallelism and domain similarities in the MAREC patent corpus. In: Salampasis, M., Larsen, B. (eds.) IRFC 2012. LNCS, vol. 7356, pp. 12–27. Springer, Heidelberg (2012)
[12] Koster, H.-A.C., Beney, J., Verberne, S., Vogel, M.: Phrase-Based Documentation Categorization. In: Croft, W.B., Lupu, M., Mayer, K., Tait, J., Trippe, J.A. (eds.) Current Challenges in patent Information Retrieval. Springer (2011)
[13] Justeson, S.J., Katz, M.S.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1(1) (1995)
[14] Bouayad-Agha, N., Casamayor, G., Ferraro, G., Wanner, L.: Simplification of Patent Claim Sentences for their Paraphrasing and Summarization. In: Lane, H.C., Guesgen, H.W. (eds.) The 22nd International Florida Artificial Intelligence Research Society Conference, Sanibel Island, Florida, USA, May 19-21, AAAI Press (2009)
[15] Shinmori, A., Okumura, M., Marukawa, Y., Iwayama, M.: Patent claim processing for readability: structure analysis and term explanation. In: Proc. ACL-2003 Workshop on Patent Corpus Processing, Stroudsburg, PA, USA, vol. 20, pp. 56–65 (2003)
[16] Andersson, L., Mahdabi, P., Hanbury, A., Rauber, A.: Exploring patent passage retrieval using nouns phrases. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp.
676–679. Springer, Heidelberg (2013)
[17] Ramshaw, A.L., Marcu, P.M.: Text Chunking Using Transformation-Based Learning. In: 3rd Workshop on Very Large Corpora, Cambridge, MA, USA (1995)
[18] Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: Proc. of HLT-NAACL, pp. 252–259 (2003)

Concept Extraction from Patent Images Based on Recursive Hybrid Classification

Anastasia Moumtzidou, Stefanos Vrochidis, and Ioannis Kompatsiaris

CERTH-ITI, Thessaloniki, Greece
{moumtzid,stefanos,ikom}@iti.gr

Abstract. Recently, the intellectual property and information retrieval communities have shown interest in patent image analysis, which could augment the current practices of patent search by image classification and concept extraction. This article presents an approach for concept extraction from patent images, which relies upon recursive hybrid (text and visual-based) classification. To evaluate this approach, we selected a dataset from the footwear domain.

Keywords: patents, images, concepts, classification.

1 Introduction

The growing number of patent applications submitted worldwide necessitates the development of advanced patent search technologies. Recently, the Intellectual Property and the Information Retrieval communities have shown great interest in patent image search, expressed with common research activities in relevant conferences (e.g. IRFC[1], CLEF-IP[2]). Non-textual elements play a crucial role in patent search, since image examination is important to patent searchers to understand the contents and retrieve relevant patents. One of the first systems dealing with patent image search was PATSEEK [1], while more recently PatMedia [2] image search engine was developed. Following the recent challenges in image analysis (semantic indexing, semantic gap), the latest approaches in patent image search deal with patent image classification [3] and concept extraction [4]. However, the motivation behind the interest in patent concept-based search is also revealed by the following scenario [5]: a patent searcher searches for a dancing shoe that incorporates a rotating heel with ball bearings; at first, the patent searcher recognises the main concepts of the invention (e.g. dancing shoe) and based on them defines keywords and relevant classification areas. In many cases the important information is described with figures. Therefore, it would be important if the patent searcher could directly retrieve patents, which include figures depicting these concepts. Such concept-based retrieval functionalities could be integrated in existing patent search systems to facilitate the tasks of patent searchers.

[1] Information Retrieval Facility Conference.
(http://www.ir-facility.org/irf-conference-2012).
[2] http://www.ir-facility.org/clef-ip

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 83–86, 2013.

In this paper, we present an approach for concept extraction from patent images based on a supervised machine learning framework using Support Vector Machines (SVM) trained with textual and visual features. This work goes beyond [4] by proposing a recursive scheme for concept extraction and deals with a more complicated scenario compared the classification task of CLEF-IP 2011 (e.g. [3]), in which more generic and visually dissimilar categories (e.g. flowcharts, symbols) are considered.

2 Patent Image Concept Extraction Framework

The proposed framework (testing phase) is depicted in Figure 1. The initial step includes the extraction of all patent images and the associated captions. Then, the images are fed into the feature extraction component, where visual and textual features are generated. Subsequently, the dataset is annotated and separated into training and test set. During this step the images of the same patents are kept together. Then, three models are trained for each concept using: a) visual features, b) textual features, and c) the results of the previous models. The latter are used as features to drive a hybrid classification model to provide the final results. The test (validation) set is further split into two sets (i.e. A and B) based on the following rule: the figures with description that points to another figure of same patent (e.g. "*Fig. 2* is the front view of *Fig. 1*") belong to set B, whereas the rest to set A. This kind of descriptions is one of the main reasons for retrieving low quality results in text-based concept extraction [4]. During the testing phase (Figure 1), the figures of set A are fed into the textual and visual feature extraction components and their features are used as input to the textual and visual classifiers respectively. The final confidence score for each concept is provided by the hybrid classifier. Then, the missing parts of the descriptions of the images contained in set B are replaced by the results of the textual classifier in a recursive way. For instance, in the previous example, if "*Fig. 1*" is annotated as "ski boot", the new caption of "*Fig 2*" will be: "*Fig. 2* is the front view of the of ski boot". This process continues recursively until all the captions of the figures in set B are updated. Finally, the figures of set B are processed in a similar way with the ones of set A.

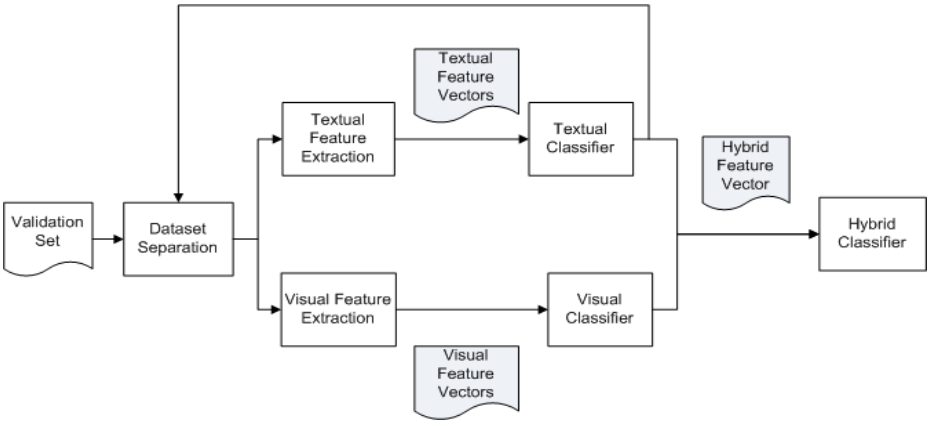


Fig. 1. Testing procedure of patent concepts

The LIBSVM [6] library was used for SVM classification. The global visual features employed are the Adaptive Hierarchical Density Histograms, which have shown discriminative power for patent images ([2], [4]). The textual feature extraction is based on the bag-of-words approach. We produce a 400-term lexicon, which includes the most frequently used words of this dataset. Indexing is performed using Lemur[3]. The weight $w_{t,d}$ of each term t for figure caption d is calculated as follows:

$$w_{t,d} = \frac{frequency\ of\ term\ t\ in\ caption\ d}{total\ number\ of\ words\ in\ caption\ d} \quad (1)$$

3 Results and Evaluation

To evaluate the approach, we use an annotated dataset (described in [4]) extracted from 355 patents. It contains around 1000 patent images depicting parts of footwear. Each image is associated with a single concept. The concepts selected are shown in Table 1, and represent specific International Patent Classification (IPC) groups (A43B21, A43B23, A43B5). Examples for ski boot concept are provided in Figure 2.

We apply three-fold cross validation at the patent level. Each training set consists of around 650 and the testing of around 250 images (in average 200 from set A and 50 from set B). Then, we evaluate the results by presenting the accuracy and F-score of the concept detectors. Table 1 contains these metrics for the validation set (A∪B). The results show that the hybrid model demonstrates better performance in terms of accuracy and F-score, while the textual features report higher F-score than the visual ones.

Table 1. Fscore and Accuracy for the validation set (A∪B)

Concept	Visual		Textual		Hybrid	
	F-score	**Accuracy**	**F-score**	**Accuracy**	**F-score**	**Accuracy**
Cleat	42,63%	88,64%	57,92%	90,58%	58,40%	89,02%
Ski boot	63,73%	92,34%	78,91%	95,55%	77,22%	95,59%
High heel	55,73%	90,50%	54,47%	89,05%	66,01%	91,96%
Heel with spring	50,63%	93,04%	51,50%	92,59%	53,24%	93,07%
Toe caps	43,66%	90,53%	74,59%	95,37%	72,45%	94,59%
Average	**51,28%**	**91,01%**	**63,48%**	**92,63%**	**65,46%**	**92,85%**

With a view to evaluating the performance of the recursive approach followed for set B, we compare the results of the proposed approach with the baseline (similar to [4]). Table 2 contains the results of both approaches, which shows that the recursive classification-based approach outperforms the baseline. Finally, Figure 2 depicts the first six figures with higher prediction scores for the concept "ski boot".

[3] Lemur Project, `http://www.lemurproject.org/`

Table 2. Performance measures F-score and accuracy for validation set B

Concept	Baseline approach		Proposed recursive approach	
	F-score	Accuracy	F-score	Accuracy
Cleat	59,95%	85,42%	77,23%	91,15%
Ski boot	57,78%	95,51%	57,78%	95,51%
High heel	71,64%	91,50%	76,23%	93,89%
Heel with spring	53,74%	91,71%	53,74%	91,71%
Toe caps	66,67%	93,46%	85,19%	97,69%
Average	61,96%	91,52%	70,03%	93,99%

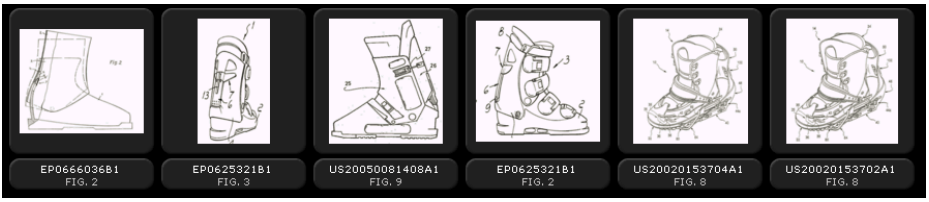


Fig. 2. Results for concept "ski boot"

4 Conclusions

In this paper, we have presented an approach for concept extraction from patent images based on recursive classification. The concept retrieval module could be a part of existing patent search systems, in order to support patent searchers in patent invalidation and valuation tasks. Future work includes testing the method by considering a larger patent database and additional concepts, in order to further test its scalability.

References

1. Tiwari, A., Bansal, V.: PATSEEK: Content Based Image Retrieval System for Patent Database. In: Proc. International Conference on Electronic Business, Beijing, China (2004)
2. Vrochidis, S., Papadopoulos, S., Moumtzidou, A., Sidiropoulos, P., Pianta, E., Kompatsiaris, I.: Towards Content-based Patent Image Retrieval; A Framework Perspective. World Patent Information Journal 32(2), 94–106 (2010)
3. Mörzinger, R., Horti, A., Thallinger, G., Bhatti, N., Hanbury, A.: Classifying patent images. In: Proceedings of CLEF 2011, Amsterdam (2011)
4. Vrochidis, S., Moumtzidou, A., Kompatsiaris, I.: Concept-based Patent Image Retrieval. World Patent Information Journal 34(4), 292–303 (2012)
5. De Marco, D.: Mechanical patent searching: a moving target. In: Patent Information Users Group (PIUG), Baltimore, USA (2010)
6. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)

Towards a Framework for Human (Manual) Information Retrieval

Fernando Loizides and George Buchanan

Abstract. Information retrieval work has mostly focused on the automatic process of filtering and retrieving documents based on a query search. The subsequesnt manual process by which the information seeker will scrutinise and triage through the retrieved documents is not thoroughly understood. Limited work, particularly for human factors in web searching have been reported on but this is usually case specific and difficult to cross reference or cross examine and compare. Furthermore, the majority of the work is also qualitatively reported on while there are no clear measures for empirically and quantitatively evaluating user behaviour and interactive systems. In this work, we introduce a universal framework which conceptualises the behavioural and procedural human process. Beyond the scholarly contribution, the framework can be employed and adapted in order for practitioners and researchers to have a foundation for evaluating both user performance and interactive systems.

1 Introduction and Motivation

Information retrieval has mostly focused on the automatic process of extracting information from a source and less on the subsequent manual process the user needs to go through. Ultimately, the information seekers will process and evaluate the returned documents and through their own personal process will filter the results even further in order to identify the useful documents. In this work we examine the manual process after the information retrieval has taken place. This process has been characterised previously as document triage[8] or document selection[25] and is the fast paced activity which occurs in order for relevance decisions to be made.

To date, researchers have not had a well defined framework with which to categorise the documents which underwent document triage. Our framework, grounded on experimental empirical data, can produce the foundation for these frameworks. We can begin to identify the relationship between document relevance and the efficiency of document triage during each stage of the process. In other words, how much '*effort*' or how many actions does it take for an information seeker to move a document from one relevance set to another? Can the time, effort and integrity of the decisions made by the users be justified as to the efficiency of the process as a whole; whether that be time or correct decisions? We can begin to identify ideal sets based on an information need and

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 87–98, 2013.

compare these to actual sets produced by the seekers. Given two different situations, which is more efficient in terms of recall and precision? These are some of the measurements which we will cover during the description of our framework.

Well established models exist that provide frameworks for information seeking (e.g. Gary Marchionini's model [17]). However, there is a need for more detailed scrutiny of more specialised activities within the information seeking process. Bates, critiques large-scale information seeking models [5] and agrees with Kuhn[14] in that "major models that are as central to a field as this one is, eventually begin to show inadequacies as testing leads to greater and greater understanding of the process being studied". Ellis recognises that "there is a need for more micro evaluation of the activities and environments of the users of information systems in order to develop an understanding of the relation of information services to those activities and environments" [12].

In this work, we move away from simply qualitative evaluations of the human factors of information retrieval. We think about triage as an activity where we have sets of documents. By doing this we are able to produce a universally understood mathematical explanation of our workings. This will give other researchers a common framework to compare our work to theirs. Furthermore, by using the theoretical stages in our model, we are able to derive measures for recall and precision at different levels of triage. These measures allow a more detailed evaluation of users' and software effectiveness during different stages of the triage process.

2 Document Triage Framework

2.1 Set Theory

Using basic set theory we can logically constrain the boundaries and limits of the document triage process using formal notions. Formal notions such as these help us achieve three purposes. Firstly, our work is directly comparable with further work from other researchers using a common mathematical language rather than simply lexical definitions. Secondly, we can refer back to these notations during model descriptions to produce a clearer understanding of actions and behaviour of users and their effects. Finally, we are able to migrate standard information retrieval measures in order to produce quantitative assessments to the human factor process.

Figure 1 shows a Venn diagram of the possible sets the documents can be located during document triage. Document triage is a part of the information seeking process. In other words, the elements of the document triage process (P) comprise a proper subset of the information seeking process (I).

$$P \subset I$$

In order for document triage to begin, a set of potentially relevant documents (C) must exist. This scenario occurs after an automatic information retrieval operation. One practical example of this is a web search engine results

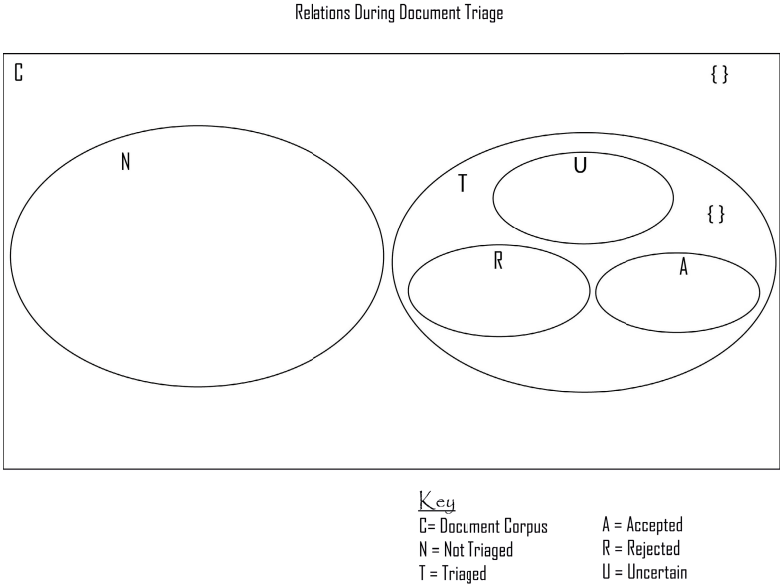


Fig. 1. Venn Diagram Describing the Possible Status of the Documents in any Given Time of the Document Triage Process

list. The resulting web page set are potentially relevant documents. When these are presented to the information seeker, the document triage process initiates. The first stage of the document triage process begins after the "execute query" stage within Marchionini's model [17]. We can also encapsulate the initiation of the triage process after the "Differentiating" stage in Ellis' information seeking model[11].

The documents may either be triaged (T) or **not** triaged (N).

$$C = \{N, T\}$$
$$C = N \cap T$$

At the beginning of the triage process all the document elements (d_i) belong to the **not** triaged set (N). The set (N) is therefore, initially equivalent to the Universal set (V).

$$\forall d_i \in N$$

At the beginning of the triage process, none of the documents are triaged. Therefore, there are no elements in the Accepted, Rejected and Uncertain sets. These sets begin empty.

$$A \cup R \cup U = \emptyset$$

There are three possible outcomes when a document is triaged (t). The information seeker either: deems the document relevant and accepts it (A), does

not deem the document relevant to their information need and rejects the document (R), or the relevance of the document remains uncertain without a decision made (U).

$$T = \{A, R, U\}$$

Sets U, R and A can never intersect.

$$U \cap R = \emptyset$$
$$U \cap A = \emptyset$$
$$R \cap A = \emptyset$$

As the triage process progresses, the documents are triaged and a decision is made on each. This stage is equivalent to Marchionini's "Examine Results" and "Extract Info" stages. This is where the majority of the document triage takes place. This process can also be seen in the "extracting" stage of Ellis' model, where an information seeker "selectively identifies relevant material"[11].

$$t : N \mapsto T$$

At *the end* of the triage process all documents are either part of: the accepted set or the rejected set. Even if a document is never reached, or if it is still in the uncertain set, at the end of the triage process it is implicitly rejected.

$$N \cup U = \emptyset$$

We can relate this to Marchionini's model's last stage of "Reflect and Stop" [17]. Marchionini's definition reflects the end of the entirety of the search process rather than one set of documents, but does include the documents being triaged. Similarly, Ellis identifies this stage as the "verifying" and "ending" stages, where the seeker will "check the accuracy of the information" and "Tie up loose ends" [12].

During the triage process, information seekers may have a change in their states of knowledge [7], causing documents to move between the accepted, rejected and uncertainty sets. However, if the information need were to change as a result of these changes of knowledge states then the current triage process ends and a new triage process begins (even if the same documents are present).

The document triage process begins after a set of documents are presented to an information seeker. The process ends when all these documents are evaluated by the information seekers, regarding their relevance to the information need. Even if documents in a results list are never reached, they are still considered to have been implicitly rejected with regards the specific triage instance. There are three stages at which a document can be evaluated. In Figure 2 we can see a model representing the document triage process and the how these stages are connected. The stages of the process are now explained in more detail.

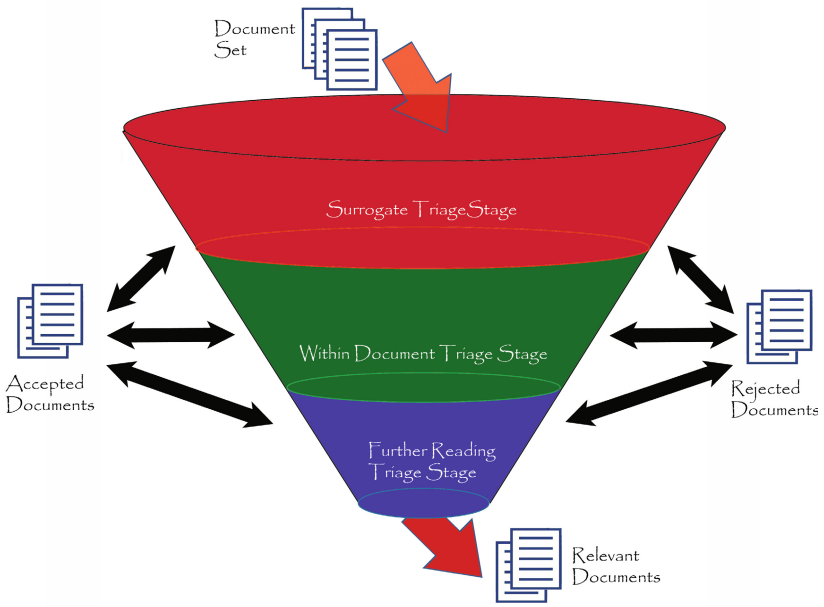


Fig. 2. High Level Abstraction Document Triage Model

2.2 The Layers of Document Triage

2.2.1 Surrogate Level Triage (Level 1)

The first contact that an information seeker is likely to have with documents, when looking for information, is at a surrogate stage (such as a results list). Often, this stage occurs when information seekers perform a query based search on a search engine. Most of the documents are either rejected or accepted at this stage. The depth of triage a user is willing to undertake when looking for information through a search results list is limited [23]. Therefore, it is common for the majority of documents to be rejected without any viewing. We also know from our previous studies that users are likely to make a relevance judgement on a document without opening the full text [8]. Users mostly have a linear approach to traversing elements at the surrogate level.

The first few seconds of an information seekers visual attention are critically important to the selection of a document for within-document triage. The evidence presented by Saracevic's early work [22] implies how decisions are unlikely to change with the addition of further information. Previous work [8, 15] gives qualitative feedback from users attesting to the fact that an information seeker's first impressions of a document, may formulate the relevance decision. This is even more so if the information need is fact finding; in other words, a short specific answer is all that is required. An example can be something like: "What is the average life span of a sheep?" It is likely that the first document selected, provided it has the answer, will be accepted and the other documents will be

rejected. [23] describes how most information seekers are not likely to view documents located further than the second page when performing triage at the surrogate level. Eye-tracking data show us visual patterns and behaviour that users engage in during a each specific triage process. Granka et al present data supporting the fact that the first result has the highest probability to be selected for further (within-document) triage [13, 21]. They continue to mention how abstracts presented at the surrogate level, ranked one and two, receive almost equal attention and reading time. "After the second link, fixation time drops off sharply". Another interesting observation is that there is a "dip around result item 6/7, both in the viewing time as well as in the number of clicks. Unlike for ranks 2 to 5, the abstracts ranked 6 to 10 receive approximately equal attention". They account this to the "fact that typically only the first 5-6 links were visible without scrolling. Once the user has started scrolling, rank becomes less of an influence for attention". A similar study was undertaken by Buscher et al, where participants' eye-gaze on contemporary search engine results was recorded [10]. Furthermore, it was hypothesized by Nicholas et al that during the surrogate triage stage, when an abstract web page was accessed before the document itself could be downloaded, users would often not continue to the download stage [19]. This was proved by Buchanan and Loizides when participants, although having the options to open a document and view the full text, chose to perform a relevance decision without proceeding to the within-document stage [16]. The same research shows how information seekers are less likely to re-triage documents even when their knowledge states change.

2.2.2 Within-Document Triage (Level 2)

The stage of within-document triage begins when the information seeker first comes into contact with the full text of an individual document. During this stage the information seeker is likely to scrutinise some parts of the document at a fast pace. There is limited in-depth reading, with skimming being the dominant behaviour. Users are searching for areas of a document that are likely to communicate the relevance of the document to their specific information need [15]. Factors which determine how extensive the within-document triage process is include: the complexity of the information need, the obscurity and layout of a document and the information seeker's triage skill.

Saracevic began to explore the extent to which titles and abstracts effect users' relevance judgements [22]. Saracevic found that relevance decisions were quite consistent when users judged based on a)title alone, b)title and abstract and c) the full text. This lay the foundation to our hypothesis that when information seekers perform triage, they make a cognitive decision early on in their triage, even if they subsequently seem to triage a document further beyond the title and the abstract. If there is no difference between decisions when users are given more information we can infer that there are problems at this level, in that the information given to the users is not being assimilated by the seekers in an effective way. Marshall and Shipman explored user behaviour during document triage [18]. They identified the concept of organizing the documents to make

sense of them, such as organising them into sets. They also began exploring how software can be used to assist users in their triage activity, by allowing the transition between the physical and the digital with the same affordances. A follow up study some 10 years later introduced how activities such as reading, browsing and the use of several pieces of software can affect as well as predict user interest [3]. By using our set theory guidelines as well as recall and precision measurements (See Section 2.3) for within-document triage, we are now able to begin investigating this further.

Interaction with documents does not necessarily need to be in an electronic mode. It may be the case however, that certain characteristics of the process done physically are adapted to the electronic environment. Buchanan and Loizides investigated the differences and similarities between participants performing document triage using physical and electronic means[16]. Evidence was given to support that users do categorize documents into three or more sets when performing triage in a physical environment, but they are constrained to using either their memory or an electronic note taking piece of software. The framework presented in this paper can be used to represent both physical as well as digital modes.

To date, different types of users' navigational patterns have been reported on [2, 20]. The results produced by those studies include documents that were being read rather than triaged. Some generic results showed how "navigation was found to be irritatingly slow and distracting" while others report that "assessing document length was difficult to do in any incidental or implicit way" [20]. It was not until later that an empirical study of navigational patterns during the triage process was performed [15]. This study identified four dominant navigational patterns. Using a tailor made document reader, the researchers were able to extract more detailed information of the attention of the information seekers, regarding the document elements present on screen. It was identified again however, that simply looking at the contents on the screen was not detailed enough and a more precise means of capturing users gaze was needed.

These results were also later tested using small screen displays [16]. The same patterns of navigation and attention were also noticed and replicated on small screen displays with the only difference being a faster triage time using small screens. In all the cases of triage during within-document triage, a skimming, rather than reading behaviour is dominant.

2.2.3 Further Reading Triage (Level 3)

A stage that is often neglected and not thoroughly researched is that of triage during the further reading stage. When information seekers pass the first two levels of triage (Results List Triage and Within-Document Triage) they are left with the set of documents which are deemed to be relevant. The true relevance of the remaining documents will only likely surface after an in-depth reading process occurs. This is the point where the information seeker will actually extract the relevant information to satisfy the information need.

Information seekers decide on a perceived relevance for each document early on in their triage process. It is not until the document has been studied in-depth using active reading [1] and some information extracted that the true relevance of the document to the information need is made clear. Upon selecting a document that appears to be relevant (by using either the surrogate triage or the within-document triage methods), the information seeker will then focus to extract the necessary information from the documents selected by more focused reading rather than skimming.

To the best of our knowledge, there is no research into the triage activities that take place during this level. This phase of the document triage process is the most difficult to study qualitatively with regards to behaviour; their are limited methodological approaches that can be used and most are invasive and may breach participant anonymity.

2.2.4 Transitions

The model in Figure 2 presents the three levels of document triage. Information seekers **transition vertically** between these levels as they progress through their triage process. During these levels, documents are being judged by the information seeker in terms of relevance. These are represented as **horizontal transitions**.

Vertical Transitions. The information seeker will begin from level one, where a set of potentially relevant documents are presented and complete the triage process at the end of level three. The document triage ends only when there exist a set of documents which are not only accepted but also relevant to the information seeker's need. No further transitions exist after this takes place. During document triage the information seeker will be migrating between these levels, making the transitions bidirectional. These transitions occur when the information seeker advances his or her knowledge state while gathering information [6]. During document triage the information seeker attempts to decide the relevance of a set of documents to his or her information need. Document triage is generally a fast-paced process and the speed of decision making at each level of triage decreases.

Horizontal Transitions. As information seekers gain knowledge about the perceived contents of a document, they make relevance decisions. Equally, they may have changed their relevance decisions based on the knowledge that they gain through the material they read. As the information seeker gains knowledge into the contents of the documents he formulates a better understanding of his or her information need. Through their triage stages (vertical triage) seekers will judge documents based on their understanding of their need and also based on their evaluation of the document relevance. As the information seeker gradually gains more knowledge, the value of each document being triaged may increase or decrease. The ratings seekers will give to documents are generally tacit in that they do not explicitly rate the documents on a scale but are tentative in nature. If that rating is above a subjective threshold, which is determined by the information seeker unconsciously, then the document is no longer rejected

and moves to the **accepted documents set**. $(t : R \mapsto A)$. What contributes to the decision of an information seeker's decision on a cognitive level is not well understood. This set can be physical such as a printout or an actual set, virtual, such as an electronic bookmark, or simply a memo in someone's memory that the document is relevant. There are also documents that may be borderline, in terms of rating, or unclear as to their content. These are usually moved in an uncertainty set (this may be physical like pile or simply a mental thought of the document) and are recycled through the funnel for further triage at a later stage when more knowledge is accumulated [15]. Documents that are well under the relevant threshold are perceived as irrelevant and moved to the **rejected documents set**. These horizontal transitions are therefore a complex and important part of the model. They can be used to assess the interactions as well as the relevance rating of users. We have seen the different category sets that the documents can be classified in the previous section (also See Figure 1). The model in Figure 2 presents a higher level view of the process from start to finish and presents the state of the information seeker rather than the documents.

Horizontal transitions are bidirectional, although they are more commonly outwards away from the funnel. The reason is that it is common for documents to be accepted or rejected and placed on a set, but it is less frequent for an information seeker to re-triage a rejected or accepted document after judgement [8]. It is however, common for the seeker to gain knowledge while performing triage which will give a new perspective on previous documents triaged. When this knowledge affects the relevance threshold, and if the information needed is critical enough, then the information seeker is likely to retrieve documents from the accepted or rejected set and recycle them through part, or all of, the triage process again.

2.3 Migrating IR Metrics to Human Factors

We begin to evaluate the human IR process after a set of documents are given to a user. Marchionini describes this stage as the "Execute Query" stage [17]. The subsequent three stages in his model; namely the "Examine Results" and "Extract Info" stages, identify the human element of relevance decision making. In recent years people have been moving to interactive information retrieval and have thought about the users behaviour at the surrogate results list [24]. At each stage in the document triage process a new Recall and Precision score is produced. As information seekers progress through the stages of the triage model, they are likely to reject documents, therefore changing the values for the answer sets and the actual relevant documents within those sets.

The common Recall and Precision scores are calculated by:

$$Recall_n = \frac{|Ra_n|}{|R|} \ , \ Precision_n = \frac{|Ra_n|}{|A_n|}$$

where $|R|$ = Relevant Documents, $|Ra_1|$ = Relevant Documents in answer set, $|A|$ = Answer Set [4] and n = Stage Number.

By synthesising scores between different stages the cumulative efficiency and effectiveness of a tool (or indeed a user) can be measured. Furthermore, by analysing the scores of the same stage between different tools, we can also compare the efficiency of the different tools. We can also bring together all the measurements presented above to create a sum of efficiency across the whole document triage process. We want to optimise this as much as possible as a whole, by breaking apart the different measurements and identifying points of weakness within the triage levels and points of failure. By using the above formulas as a basis we can infer further measurements to rate the transition effectiveness between the three levels of document triage. To illustrate, we take a hypothetical scenario:

An information seeker runs a search query which returns 10,000 results. Out of these 10,000 results only 5,000 are relevant; therefore making precision at the surrogate level 0.5. The information seeker decides to proceed to the within-document triage stage but only selects 8 documents to triage at this level. If we assume that 6 of these documents are relevant then the precision at this level now becomes 0.75. During the transition between the surrogate and the within-triage stage, the information seeker has been influenced by external and internal factors to make the decisions about the documents that will be viewed in full text. These factors usually constitute the interface by which the information is presented. An empirical method for rating these information retrieval interfaces is not available. We can however, infer formulas which can give us a rating for the effectiveness of the transition. For example, if we were to calculate the change between $Precision_1$ *and* $Precision_2$ *we can then assign a score to the interface used. If we were to use the same exact document set, we can then compare and contrast different interface scores between the two levels. A similar approach is very common for assessing information retrieval algorithms in TREC [9].*

3 Summary

Our work focuses on the human manual process of information retrieval after the documents of a search query have been returned. This manual process is vital in that information seekers are ultimately the ones who will make the final decisions. This behaviour can vary depending on the individual seeker's circumstances and can be influenced by several variables such as tool availability, time and experience. These concepts have thus far not been well understood or universally modelled. In this paper we have introduced a mathematical common framework conceptualising the document triage process. By using this assessment we can now evaluate users and tools at different parts of the triage process. Our model and mathematical framework ties together to produce measures that can also be used by others to compare users, software tools and methods. Future work includes testing the framework on individual case scenarios as well as expanding the granularity of the model's parts.

References

[1] Adler, M.J., Van Doren, C.L.: How to Read a Book. Simon and Schuster (1996)
[2] Alexander, J., Cockburn, A.: An empirical characterisation of electronic document navigation. In: Graphics Interface, pp. 123–130 (2008)
[3] Badi, R., Bae, S., Michael Moore, J., Meintanis, K., Zacchi, A., Hsieh, H., Shipman, F., Marshall, C.C.: Recognizing user interest and document value from reading and organizing activities in document triage. In: IUI 2006: Proceedings of the 11th International Conference on Intelligent user Interfaces, pp. 218–225 (2006)
[4] Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
[5] Marcia Bates, J.: The design of browsing and berrypicking techniques for the online search interface. Online Information Review, 407–424 (1989)
[6] Belkin, N.J.: Anomalous states of knowledge as a basis for information retrieval. Canadian Journal of Information Science (5), 133–143 (1980)
[7] Belkin, N.J., Oddy, R.N., Brooks, H.M.: Ask for information retrieval: Part i. background and theory. The Journal of Documentation 2, 61–71 (1982)
[8] Buchanan, G., Loizides, F.: Investigating document triage on paper and electronic media. In: Kovács, L., Fuhr, N., Meghini, C. (eds.) ECDL 2007. LNCS, vol. 4675, pp. 416–427. Springer, Heidelberg (2007)
[9] Buckley, C., Salton, G., Allan, J., Singhal, A.: TREC (1994)
[10] Buscher, G., Dumais, S.T., Cutrell, E.: The good, the bad, and the random: an eye-tracking study of ad quality in web search. In: SIGIR, pp. 42–49 (2010)
[11] Ellis, D.: A behavioral approach to information retrieval system design. Journal of Documentation, 171–212 (1989)
[12] Ellis, D.: The effectiveness of information retrieval systems: the need for improved explanatory frameworks. In: Social Science Information Studies, pp. 261–272 (1984)
[13] Granka, L.A., Joachims, T., Gay, G.: Eye-tracking analysis of user behavior in www search. In: SIGIR, pp. 478–479 (2004)
[14] Thomas Khun, S.: The Structure of scientific revolutions, 2nd edn. University of Chicago Press (1970)
[15] Loizides, F., Buchanan, G.: An empirical study of user navigation during document triage. In: Agosti, M., Borbinha, J., Kapidakis, S., Papatheodorou, C., Tsakonas, G. (eds.) ECDL 2009. LNCS, vol. 5714, pp. 138–149. Springer, Heidelberg (2009)
[16] Loizides, F., Buchanan, G.R.: Performing document triage on small screen devices. part 1: structured documents. In: Proceeding of the Third Symposium on Information Interaction in Context, IIiX 2010, pp. 341–346. ACM, New York (2010)
[17] Marchionini, G.: Information Seeking in Electronic Environments. Cambridge University Press (1995)
[18] Marshall, C.C., Shipman III, F.M.: Spatial hypertext and the practice of information triage. In: Proceedings of the Eighth ACM Conference on Hypertext, pp. 124–133 (1997)
[19] Nicholas, D., Huntington, P., Jamali, H.R., Watkinson, A.: The information seeking behaviour of the users of digital scholarly journals, pp. 1345–1365 (2006)
[20] O'Hara, K., Sellen, A.: A comparison of reading paper and on-line documents. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 335–342 (1997)

[21] Pan, B., Hembrooke, H., Gay, G., Granka, L.A., Feusner, M.K., Newman, J.K.: The determinants of web page viewing behavior: an eye-tracking study. In: ETRA, pp. 147–154 (2004)
[22] Saracevic, T.: Comparative effects of titles, abstracts and full text on relevance judgments. Journal of the American Society for Inf. Science (22), 126–139 (1969)
[23] Spink, A., Jansen, B.J., Wolfram, D., Saracevic, T.: From e-sex to e-commerce: Web search changes. Computer 35, 107–109 (2002)
[24] Voorhees, E.M., Harman, D.K.: TREC: Experiment and Evaluation in Information Retrieval. MIT Press (2005)
[25] Wang, P., Soergel, D.: A cognitive model of document use during a research project. study i. document selection. Journal of the American Society for Information Science 49(2), 115–133 (1998)

A Generalized Framework for Integrated Professional Search Systems

Michail Salampasis and Allan Hanbury

Institute of Software Technology and Interactive Systems,
Vienna University of Technology, Austria
{salampasis,hanbury}@ifs.tuwien.ac.at

Abstract. This paper presents a framework for Integrated Professional Search (IPS) systems. The framework provides a context to better classify and characterize what IPS systems are, but it is also used to better understand the design space of IPS systems. The framework suggests an architecture and methodology to build loosely coupled IPS systems in which each of their search tools have little or no knowledge of the details of other search tools or components. The paper also describes, as a case study of using the proposed framework, the architecture and the main functionalities of a patent search system and a medical search system. The integration of different search tools into these search systems is discussed to demonstrate the flexibility of the framework that facilitates external search tools to be integrated into the search systems.

1 Introduction

Searching for information in large scale datasets has become one of the most pervasive and powerful applications of computing nowadays. Web search engines have proved extremely effective and efficient using the "query box" paradigm and ranked lists of search results to find relevant information for general purpose retrieval tasks. To a large extent this has led to the great success and exponential growth of the Web. Another important implication is that the success of web search engines has produced a generation of digital natives that have grown up with instant access to information and for whom the first 10 hits in Google are the truth [1].

On the other hand search technologies are used for **professional search** (e.g. patent, medical, engineering, scientific literature search) for more than 40 years as an important method for information access [2]. Despite the tremendous success of web search technologies, there is a significant skepticism from professional searchers and a very conservative attitude towards adopting search methods, tools and technologies beyond the ones which dominate their domain [3]. An example is patent search where professional search experts typically use the Boolean search syntax and quite complex intellectual classification schemes [4]. Of course there are good reasons for this. A patent search professional often carries out search tasks for which high recall is important. Additionally s/he would like to be able to reason about how the results have been produced, the effect of any query re-formulation action in getting a new set of

M. Lupu, E. Kanoulas, and F. Loizides (Eds.): IRFC 2013, LNCS 8201, pp. 99–110, 2013.

results, or how the results of a set of query submission actions can be easily and accurately reproduced on a different occasion (the latter is particularly important if the patent searcher is required to prove the sufficiency of the search in court at a later stage). Classification schemes and metadata are heavily used because it is widely recognized that once the work of assigning patent documents into classification schemes is done, the search can be more efficient and language independent.

Generally speaking, apart from the differences in the patent search domain presented above, there are a number of important parameters and characteristics that differentiate professional search (that is search in the workplace or search for a professional reason or aim) from web search such as: lengthy search sessions (even days) which may be suspended and resumed, the notion of relevance can be different, many different sources will be searched separately, and focus is on specific domain knowledge in contrast to public search engines which are not focused on expert knowledge.

The current trend in professional search is towards Integrated Professional Search Systems [5][6][7][8]. Although it is relatively easy to differentiate professional search from 'public search' with a number of characteristics (some of them briefly outlined above), the concept of an *integrated* search system is not clear. Most definitions found in the Information Retrieval (IR) literature converge to use the term "integrated" to define search systems that simultaneously access a number of different data sources providing a single point of search. This view is much more compatible with the Federated Search view that allows the simultaneous search of multiple resources.

One objective of this paper is to present a general framework for studying integrated professional search systems, therefore we should make clear from the beginning that in our framework and definition of *integrated professional search systems,* the term integrated is used beyond the way that it is used in Federated (or aggregated) search. It is primarily used as a method for integrating multiple search tools that can be used (in parallel or in a pipeline) from the professional searcher during a potentially lengthy search session. Some of these tools may serve for example the need of retrieving information from distributed data sets as it happens in federated search, or to expand and suggest a query but other tools may operate at runtime to deliver to the searcher's desktop multiple information flows and views. As a result our definition of integrated professional search systems primarily describes a rich information seeking environment for different types of searches, utilizing multiple search tools and exploiting a diverse set of IR and Natural Language Processing (NLP) technologies.

The main idea behind the framework which is presented in this paper, namely Electra, builds upon this idea of integrated professional search systems and has four functional modules a) information and information sources, b) metadata and semantics, c) session and d) information views and user interaction. The main objectives of the framework are: a) provide an analytic method to study and understand the design space of professional search systems, b) classify different IR and NLP technologies according to the functionality and services they provide to different modules, c) describe and compare professional search systems in a more systematic and independent way and, d) provide an architecture for developing interoperable search systems based on a set of cooperating IR/NLP tools.

Section 2 presents the framework for professional search systems, the rationale behind defining the four basic modules. Section 3 shows another use of the framework which is increasing the understanding of the design space for professional search systems and what are the core protocols that should exist to integrate multiple IR technologies within an integrated search system. Section 4 presents a case study of a patent search system and a medical search system and uses the framework terminology to analyze the systems and the tools integrated. Section 5 concludes the paper.

2 A Framework for Integrated Professional Search Systems

Figure 1 presents the most widely used model for search systems. A user driven by an information need constructs a query in some query language. The query is submitted to a system that selects from a collection of documents (corpus) which are already indexed, those documents that match the query using certain rules of the retrieval engine. A query refinement process might be used to create new queries and/or to refine the results. More or less traditional search systems were based in this basic model and web search is also based on a modification of this model [9].

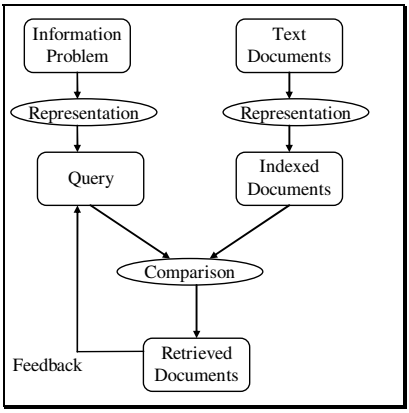


Fig. 1. Basic processes of traditional IR systems

Despite the fact that many different IR and/or NLP technologies are used in the various sub-processes depicted in Figure 1, and many exciting developments have been achieved that increased the efficiency and the effectiveness of this model, from an architectural point of view, it is important to observe that the relationships and dependencies between the different technologies, the core services which are used and the workflows and interactions which are executed in a search system during an information seeking process are not well defined.

For example, many search systems today combine a faceted search module based on static or dynamically extracted metadata. The faceted search tool and views can be combined with the "traditional" ranked result list. This simple and very common design of combining multiple search views is not captured in the basic IR model presented in

Figure 1. We believe this is an important drawback. The IR and NLP research communities have achieved tremendous progress in developing new algorithms and tools in various areas of information processing and retrieval, however there was little attention paid on how these results can come together to design next generation search systems. This view is supported by the fact that using and managing information workflows between autonomous (and possibly distributed) IR or NLP tools/services is the main design method used by different groups working in managing languages resources [10] or professional search systems [11].

The Electra[1] framework aims to address this problem and has as its primary goal to make explicit four functional modules in professional search systems and provide an underlying architecture based on which different IR and NLP technologies can coexist and tools interoperate providing a set of search services in an integrated and uniform way. The four functional modules of the framework are shown in Figure 2. The bordered areas around the functional modules depict the communication or other protocols which may be used between tools residing in different modules.

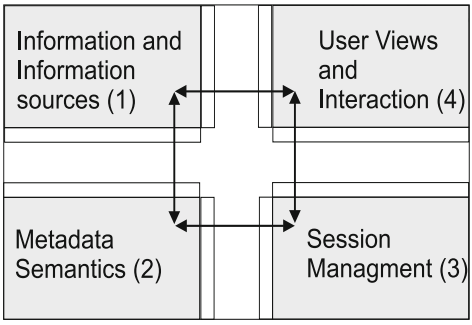


Fig. 2. Electra: a framework for Professional Search Systems

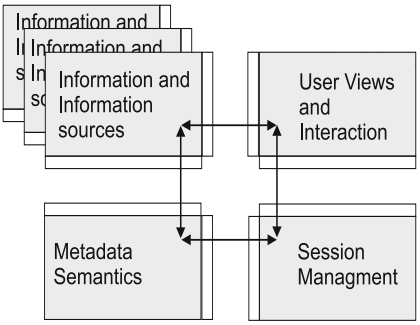


Fig. 3. A Federated Professional Search System searching multiple remote source as depicted using the Electra Framework

Functional module 1 (Information and Information Sources) is where all primitive information in different modalities (text, image, audio etc.) is stored. Information can be organized in a single collection (logically or physically) as it happens in most search systems, or it can be found in many different collections which are physically distributed in different servers. For example Figure 3 shows an instantiation of the framework that depicts a federated search system approach where multiple information sources exist and are queried by the searcher through automated source selection and results merging services[2].

The framework captures the traditional IR model (illustrated in Figure 1) as this model is already encapsulated in the functional modules 1 (Information Sources) and

[1] The name Electra comes from the hotel where an interdisciplinary working group meeting on Integrating IR technologies took place and the framework was first presented. The framework is also inspired by the flag taxonomy for Open Hypermedia Systems [22].

[2] At this high level of using the Electra framework we do not illustrate these services as there is not enough space to clearly show them in the figure.

4 (View-Interact). However the framework additionally offers a much wider range of design space capable of capturing more comprehensively the extensive range of research and development activities of modern IR and NLP tools and systems.

For example functional module 2 (Metadata-Semantics) explicitly captures and incorporates into the design space of next generation professional search systems the importance of the so-called Knowledge Extraction and Organization, e.g. classification schemes, taxonomies, ontologies. These are important prerequisites and resources for developing intelligent search tools and search systems that no longer just do what the professional searcher says but also what he means [12]. This explicit distinction between raw information and metadata/semantics signifies the importance of IR and NLP technologies such as entity mining and extraction, faceted and semantic search and generally methods seeking to improve search accuracy by understanding the contextual meaning of terms to generate more relevant results.

The right section of the framework, i.e. functional modules 3 and 4 are concerned with the runtime and user aspects of the information seeking process. What is important to observe in the framework is that having functional module 3 (Session) which is fully separated from the more static left part of the framework (information and metadata) has the important implication that instantiations of information and metadata during a session can be treated and managed separately from their original sources, and therefore can be stored as first-class objects. This means that the session data produced during a search process can become information itself which can be stored, searched, processed and analyzed. An important implication for search systems developing this module of the framework is that they can manage and store session data as first-class objects and therefore increase the reproducibility of a search process and preserve complete statefull sessions that can be stored and managed at a later stage. This is a very important requirement for professional search systems.

The last functional module of the Electra framework is View/Interact. The main innovative feature the framework suggests is the potentially parallel coordinated use of multiple views produced from various search services accessing the data source(s) under examination. These views can be a "simple" ranked list of documents produced out of a retrieval algorithm aiming to deliver the "best" 10 results, but other views may be produced as a result of combining or filtering information (using Linked Open Data for example) or using metadata (e.g. using faceted search based on already produced or dynamically extracted entities).

It is important to mention that the framework can be used in different levels of abstraction in order to study and classify a professional search system, or compare a number of different search systems. However, there is a need to define a clear communication and coordination architecture before the framework can be useful as a complete architecture and a starting point for the development of an integrated professional search system. The framework addresses this opportunity for becoming an underlying platform for designing and developing professional search systems by explicitly stating the existence of protocols between the four functional modules. These protocols can take various forms based on the work that has been produced in the past or relatively recent in various fields of computing such as workflow management [13], multi-agent systems [14], agent-based software engineering [15], service oriented computing [16] and web services [17].

3 The Framework as a Tool for Designing IPS Systems

The complexity of the tasks which need to be performed by professional searchers, which usually include not only retrieval but also information analysis and monitoring tasks, require association, pipelining and possibly integration of information as well as coordination of multiple and potentially concurrent search views produced from different datasets and search tools. Many facets of IR/NLP technology (e.g. exploratory search, federated search, IR over query sessions, cognitive IR approaches, Human Computer and Information Retrieval) aim to at least partially address these demands. However, generally all of this relevant research, which is vitally important for the development of next generation search systems, is fragmented and usually the main concern is at the algorithmic level and not so much on the process of integrating a tool within a professional search system.

The Electra framework can be a step towards addressing this need. To evaluate the applicability of the Electra framework we used it during a research networking meeting, where 38 IR/NLP scientists and professionals organised groups participated, in an experiment based on the living lab concept. The main purpose was to evaluate the expressiveness of the Electra framework within the context of four different groups: 1) Language Resources and Processing Infrastructures for IR/NLP, 2) User Centred Aspects of IR/NLP, 3) Semantic Search and Faceted Search and 4) Visualization, Distributed and Social Search).

The participants of the meeting attended a short presentation about the framework and afterwards they were asked to attend break-out sessions organized for each group. In each break-out session each group was provided a flip-chart paper depicting the framework and post-it notes in four different colors, with each color representing IR/NLP tools, IR/NLP technologies, concepts and core services. The task of each group was to interactively discuss the framework within the context the topic of the group and allocate (using a colored post-it) in the topology map of the framework the key IR/NLP technologies, concepts, tools and core services. The break-out sessions lasted 45 minutes. The results are presented in Figure 4. Each color represents a different aspect (yellow: IR/NLP technologies, cyan: concepts, green: core services, orange: tools) as they were placed in the framework's topology by the members of each group. After the break-out sessions a plenary session followed where each group presented the design map it had produced.

One first outcome of the trial described above is that the Electra framework could be used in the design process of defining the technologies, concepts, tools, and core services (as well as components and data) for a search system to satisfy the requirements of a specific design. Although the participants received very few training about the framework and how to use it, they managed to produce reliable and expressive "design maps" of search systems that use the specific technologies about which each group was mostly knowledgeable and focused. Of course it must be said that the the "design maps" were produced at a high level of abstraction. However, the framework can be used at lower level of granularity to facilitate more detailed design of an IPS. Obviously to validate this more extensive running experiments are required conforming more accurately to the living lab concept which may last weeks or even months.

Generally we deem that the framework should not be seen within the strict context of IR evaluation tradition, although several aspects of the framework can be quantified.

For example the number and the distribution of different colors (aspects of the design) in the four different functional modules indicate the view and the focus of a group towards specific aspects of the development of an IPS. Overlap between the various components (i.e. technologies, concepts, tools, core services) and the overall topology produced (i.e. where these components were placed from each group) is an indicator of the common understanding (or not) of the various components that should come together and utilized to meet specific requirements or user needs.

We cannot provide a detailed analysis of the outcomes of the group discussions in this paper, but the preliminary study we presented here indicates the following:

- The framework helps IR/NLP researchers and designers of search systems to understand the design space of integrated professional search systems.
- The large number of concepts, tools, services and IR/NLP technologies that have been identified reveals the broad spectrum of different IR/NLP disciplines that can be involved in the development of next generation professional search systems.
- The framework could be used, at different levels of abstraction, as a tool to organize the design process of an IPS requiring collaboration between different groups.

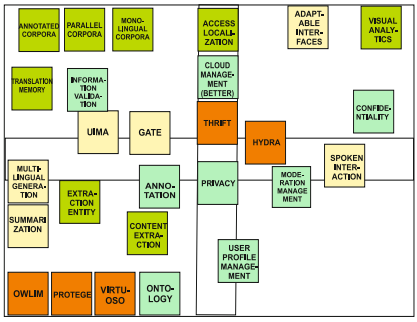


Fig. 4. The topology of the framework as it was produced by group 1 (Language Resources and Processing Infrastructures)

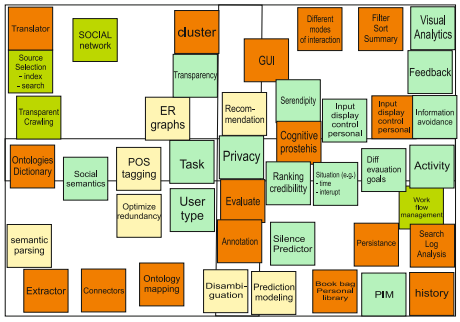


Fig. 5. The topology of the framework as it was produced by group 2 (User Centred Aspects)

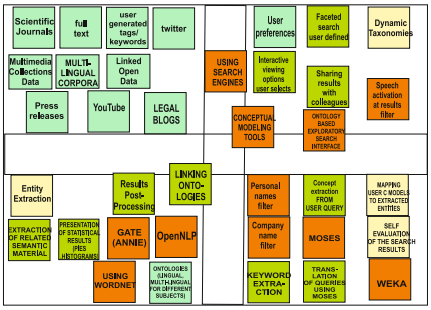


Fig. 6. The topology of the framework as it was produced by group 3 (Semantic Search and Faceted Search)

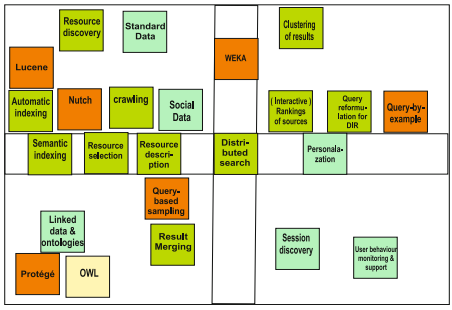


Fig. 7. The topology of the framework as it was produced by group 4 (Visualization and Distributed and Social Search)

4 Case Study: Using the Framework to Classify IPS Systems

The **PerFedPat**[3] system is a federated patent search system based on ezDL [18], [19] a system for developing interactive search applications. PerFedPat provides core services and operations for being able to search, using a federated method, multiple online patent resources (currently Esp@cenet, Google patents, Patentscope and the CLEF-IP collection), thus providing access to multiple patent sources while hiding complexity from the end user who uses a common query tool for querying all patent datasets at the same time. Wrappers are used which convert PerFedPat's internal query model into the queries that each remote service can accept. "Translated" queries are routed to remote search systems and their returned results are internally re-ranked and merged as a single list presented to the patent searcher. Except patent resources there are other resources already supported by ezDL, most of them offering access to online bibliographic search services. Based on this architecture PerFedPat aims to become a pluggable system which puts together the following components: retrieval, selection, integration, presentation and adaptation (Figure 8).

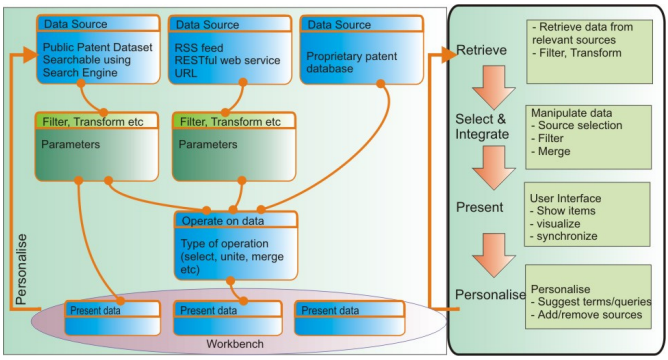


Fig. 8. PerFedPat architecture and component overview

One innovative feature of PerFedPat is that it enables the use of multiple search tools which are integrated in PerFedPat. The tools that a designer will decide to integrate into a patent search system, do not only have to do with existing IR technologies, but probably more with the context in which a patent search is conducted and the professional searcher's attitude. Furthermore, it is also very important to understand a search process and how a specific tool can attain a specific objective of this process and therefore increase its efficiency.

Currently the search tools which are integrated are a) an International Patent Classification (IPC) selection tool b) a tool for faceted search producing different facets of the results retrieved based on existing metadata in patents, c) a tool producing clustered views of patent search results d) a MT tool for translating queries. The first tool aims to support a specific objective during prior art search, i.e. to narrow the search by identifying relevant IPC codes and the effectiveness of the IPC selection tool

[3] www.perfedpat.eu

method has been evaluated [20]. In this paper we discuss more on the approach which have been used to integrate this and the other tools in PerFedPat.

From the perspective of the Electra framework what is also important to present is that the integration of the IPC suggestion tool was implemented sending *http requests* to an external server providing the IPC selection services. The server hosting the IPC selection tool receives the requests and sends a response back about the IPC codes suggested. It is important also to mention that for the IPC tool to operate there is a need to access certain metadata that are produced using Distributed IR core services (resource representation) and are managed locally by the IPC selection server. This data could also exist in the original PerFedPat server and could be sent on request to different tools which need to access such data. It is therefore important to mention that interoperability at the process level is achieved, however this process level interoperability is not based on full exchange of metadata but some form of regular updates may be necessary.

In PerFedPat there are more search tools integrated in similar way (tools for faceted search, clustered views of results). From an information seeking process perspective, the integration of different search tools in addition to the basic ranked list of patent documents returned from the DIR retrieval engine, allows different views of patent information to coexist.

This process-oriented integration provides some useful services to the PerFedPat patent search system but synchronization between the tools is required so one event or action in one tool can update the views produced from the other tools. For example selecting an IPC code may affect the results presented in the faceted search tool.

The Khresmoi project[4] is building an integrated search system for multilingual, multimodal health and medical information. Like PerFedPat, the Khresmoi search system is based on a Service-Oriented Architecture (SOA), using a Service Component Architecture (SCA) model. The Khresmoi system provides the ability to search both text and images, but only the text components will be considered here. The text search components for *Khresmoi Professional*, the search engine instantiation aimed at medical professionals, is shown in Figure 9.

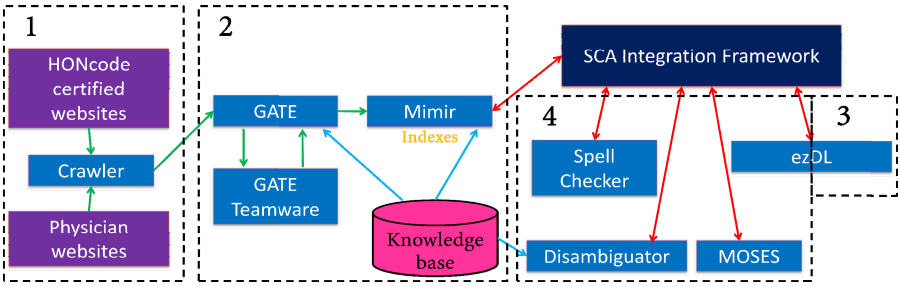


Fig. 9. Khresmoi text search Service-Oriented Architecture. The dotted lines represent the functional modules of the Electra framework.

[4] http://khresmoi.eu

As for PerFedPat, ezDL is used as the search interface for Khresmoi Professional. Khresmoi is based to a large extent on open source software, including GATE, the General Architecture for Text Engineering, and Mimir, a search engine that makes use of GATE text annotations. Furthermore, it makes use of the MOSES machine translation software trained on medical documents to translate both queries and document sections. Further components such as a spell checker and disambiguator for queries are called. A knowledge base supports the annotation and disambiguation components.

We now briefly analyze Khresmoi in terms of the Electra framework. For this specialized health and medical search engine, attention needs to be paid to the resources indexed (functional module 1), to ensure that they are of sufficient quality. We chose to index websites certified by the Health on the Net foundation [21], as well as websites manually selected as being important to physicians. For the metadata (functional module 2), there exist many medical ontologies and vocabularies. For this reason we settled on the Linked Life Data, which fuses and cross-links over 30 biomedical knowledge resources. The GATE and Mimir tools make use of the metadata by annotating the texts, and searching based partly on the annotations. Through the use of the ezDL framework, session management (functional module 3) is included in the system, which stores a detailed query log and allows users to place documents in their personal library, tag them and share them with other users. Finally, the user can interact with the information (functional module 4) by machine translating it through transparent calls to the MOSES service, or using tools provided by ezDL such as filtering results through the use of facets or search within the results, or generating summaries in the form of word clouds. As for PerFedPat, it is important to note that the integration was done by making all components available through web service interfaces and defining data exchange protocols between the web services.

5 Conclusion

The Electra framework provides a method to classify integrated professional search systems. It can be used as a method to identify the main functionalities and characterise them in a relatively system independent way. For example from studying the framework produced for the PerFedPat system it is relatively easy to identify the basic components of the system and characterise it as a Federated Patent Search system which integrates other semantic search and DIR tools which interoperate during a search process. The framework also provides a useful topology for better understanding the design space of professional search systems and how different IR/NLP technologies can be integrated to enable rich information seeking environments where different tools can support specific objectives. We expect that the taxonomic diagrams the framework offers could be used as a starting point for designing professional search systems presented in Section 4.

In Section 4 we presented, within the context of the framework, a description of two professional search systems, PerFedPat for patent search and Khresmoi for medical search. Both descriptions are heavily based on the concepts already provided by

the Electra framework. Understanding such complex professional search systems with less effort is an important objective which we believe is quite important for the community involved in the development of integrated search systems and IR/NLP technologies in general. To that end the framework can be a useful tool to increase the understanding between the different disciplines involved in the development of next generation systems, and to harmonise R&D in this area. Also, it may be useful to offer a common vocabulary for the design of professional search systems.

The fact that the framework, apart from the functional modules, focuses also on protocols between the tools found in different functional modules shows the practical aspect of the framework towards developing real professional search systems applications. We believe that the development of integrated professional search systems will greatly benefit if communication and coordination protocols are well defined to allow interoperability between different IR and NLP tools in a service oriented paradigm. The exact methodology (message exchange, process oriented, workflow management, etc.) which will be used to develop such protocols remains an open issue, probably the task of a standardization activity. However we believe that it is an interesting way to develop search systems but it can also become an attractive business model for research groups building different types of IR/NLP technologies and tools or for SMEs developing search solutions.

Of course we do not ignore the major concern of the communication overhead and the other efficiency costs which occur in the development of search systems based on interoperable components. However, we believe that the scalability which has been practically demonstrated over the last ten years, together with carefully crafted protocols and workflows, indicates that this is a feasible path towards the design of next generation professionals search systems. In fact we consider this movement of "Open Search Tools" as a paradigm which can couple to the movement of Open Data and could eventually lead to Information Seeking Environments which will emphasise information finding and understanding rather than only information retrieval.

Acknowledgements. The research leading to these results has received funding from the European Un-ion Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 27552 (PerFedPat) and n° 257528 (KHRESMOI).

References

1. Lagemaat, W.G.: The future of information tools and technology – Our joint effort. World Patent Information 35(2), 93–94 (2013)
2. Adams, S.: The text, the full text and nothing but the text: Part 1 – Standards for creating textual information in patent documents and general search implications. World Patent Information 32(1), 22–29 (2010)
3. Krier, M., Zaccà, F.: Automatic categorisation applications at the European patent office. World Patent Information 24(3), 187–196 (2002)
4. Dirnberger, D.: A guide to efficient keyword, sequence and classification search strategies for biopharmaceutical drug-centric patent landscape searches - A human recombinant insulin patent landscape case study. World Patent Information 33(2), 128–143 (2011)

5. Kohn, A., Bry, F., Manta, A.: Professional Search: Requirements, Prototype and Preliminary Experience Report. In: Iadis International Conference WWW (2008)
6. Lund, H., Lauridsen, H., Hansen, J.H.: Summa – integrated search. Zuletzt Abgerufen 13, 1–13 (2010)
7. Masiakowski, P., Wang, S.: Integration of software tools in patent analysis. World Patent Information 35(2), 97–104 (2013)
8. Salampasis, M., Fuhr, N., Hanbury, A., Lupu, M., Larsen, B., Strindberg, H.: Integrating IR Technologies for Professional Search. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 882–885. Springer, Heidelberg (2013)
9. Broder, A.: A taxonomy of web search. ACM Sigir Forum 36(2), 3–10 (2002)
10. Bel, N.: Platform for Automatic, Normalized Annotation and Cost- Effective Acquisition of Language Resources for Human Language Technologies. PANACEA. Procesamiento del Lenguaje Natural 45, 327–328 (2010)
11. Hanbury, A., Müller, H.: Khresmoi – multimodal multilingual medical information search. In: MIE Village of the Future (2012)
12. Wolter, B.: It takes all kinds to make a world – Some thoughts on the use of classification in patent searching. World Patent Information 34(1), 8–18 (2012)
13. Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P., Bhagat, J., Belhajjame, K., Bacall, F., Hardisty, A., Nieva de la Hidalga, A., Balcazar Vargas, M.P., Sufi, S., Goble, C.: The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Research, p. gkt328– (May 2013)
14. Salampasis, M., Tait, J., Hardy, C.: An Agent-Based Hypermedia Framework for Designing and Developing Digital Libraries, p. 5 (May 1996)
15. Sharma, D., Ma, W., Tran, D., Anderson, M.: A Novel Approach to Programming: Agent Based Software Engineering. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4253, pp. 1184–1191. Springer, Heidelberg (2006)
16. Papazoglou, M.P., Georgakopoulos, D.: Service-Oriented Computing. Communications of the ACM 46(10), 24–28 (2003)
17. Huhns, M.N.: Agents as Web services. IEEE Educational Activities Department 6(4), 93–95 (2002)
18. Fuhr, N., Klas, C.-P., Schaefer, A., Mutschke, P.: Daffodil: An Integrated Desktop for Supporting High-Level Search Activities in Federated Digital Libraries. In: Agosti, M., Thanos, C. (eds.) ECDL 2002. LNCS, vol. 2458, pp. 597–612. Springer, Heidelberg (2002)
19. Fuhr, N.: An Infrastructure for Supporting the Evaluation of Interactive Information Retrieval, p. 4503 (October 2011)
20. Salampasis, M., Paltoglou, G., Giahanou, A.: Report on the CLEF-IP 2012 Experiments: Search of Topically Organized Patents. In: CLEF (Online Working Notes/Labs/Workshop) (2012)
21. Boyer, C., Baujard, V., Geissbuhler, A.: Evolution of Health Web certification through the HONcode experience. Studies in Health Technology and Informatics 169(6), 53–57 (2011)
22. Østerbye, K., Wiil, U.: The flag taxonomy of open hypermedia systems. In: Proceedings of the Seventh ACM Conference on Hypertext (1996)

Author Index