Carol Peters Giorgio Maria Di Nunzio
Mikko Kurimo Thomas Mandl
Djamel Mostefa Anselmo Peñas
Giovanna Roda (Eds.)

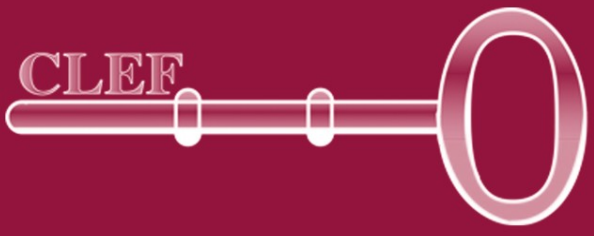
LNCS 6241

Multilingual Information Access Evaluation I

Text Retrieval Experiments

10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009
Corfu, Greece, September\October 2009
Revised Selected Papers, Part I

1 Part I



Springer

Lecture Notes in Computer Science 6241

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

Editorial Board

Carol Peters Giorgio Maria Di Nunzio
Mikko Kurimo Thomas Mandl Djamel Mostefa
Anselmo Peñas Giovanna Roda (Eds.)

Multilingual Information Access Evaluation I

Text Retrieval Experiments

10th Workshop of the Cross-Language Evaluation Forum,
CLEF 2009
Corfu, Greece, September 30 - October 2, 2009
Revised Selected Papers

Springer

Volume Editors

Carol Peters
ISTI-CNR, Area Ricerca CNR
56124 Pisa, Italy
E-mail: carol.peters@isti.cnr.it

Giorgio Maria Di Nunzio
University of Padua
35131 Padova, Italy
E-mail: dinunzio@dei.unipd.it

Mikko Kurimo
Aalto University
00076 Aalto, Finland
E-mail: mikko.kurimo@tkk.fi

Thomas Mandl
University of Hildesheim
31141 Hildesheim, Germany
E-mail: mandl@uni-hildesheim.de

Djamel Mostefa
ELDA/ELRA, 75013 Paris, France
E-mail: mostefa@elda.org

Anselmo Peñas
LSI-UNED, 28040 Madrid, Spain
E-mail: anselmo@lsi.uned.es

Giovanna Roda
Matrixware, 1060 Vienna, Austria
E-mail: giovanna.roda@gmail.com

Managing Editors

Pamela Forner and
Danilo Giampiccolo
CELCT, Trento, Italy
Email: {forner; giampiccolo}@celct.it

Library of Congress Control Number: 2010934130

CR Subject Classification (1998): I.2.7, H.3, H.4, H.2, H.5, I.7

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

ISSN 0302-9743
ISBN-10 3-642-15753-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15753-0 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The tenth campaign of the Cross Language Evaluation Forum (CLEF) for European languages was held from January to September 2009. There were eight main evaluation tracks in CLEF 2009 plus a pilot task. The aim, as usual, was to test the performance of a wide range of multilingual information access (MLIA) systems or system components. This year, about 150 groups, mainly but not only from academia, registered to participate in the campaign. Most of the groups were from Europe but there was also a good contingent from North America and Asia.

The results were presented at a two-and-a-half day workshop held in Corfu, Greece, September 30 to October 2, 2009, in conjunction with the European Conference on Digital Libraries. The workshop, attended by 160 researchers and system developers, provided the opportunity for all the groups that had participated in the evaluation campaign to get together, compare approaches and exchange ideas.

The schedule was divided between plenary track overviews, and parallel, poster and breakout sessions presenting the CLEF 2009 experiments and discussing ideas for the future. There were several invited talks. Noriko Kando, National Institute of Informatics, Tokyo, reported on the evolution of NTCIR (NTCIR is an evaluation initiative focussed on testing information access technologies for Asian languages), and Jaap Kamps of the University of Amsterdam presented the main outcomes of a SIGIR workshop on the "Future of IR Evaluation." In the final session, Donna Harman, US National Institute of Standards and Technology, summed up what she felt were the main achievements of CLEF over these ten years of activity. The presentations given at the CLEF workshop can be found on the CLEF website at www.clef-campaign.org.

The workshop was preceded by two related events. On September 29, a one-day Workshop on Visual Information Retrieval Evaluation was held. This workshop was sponsored by the THESEUS program and co-organized by the Fraunhofer Institute for Digital Media Technology. The participants discussed the results of the ImageCLEF initiative and identified new challenging image retrieval and analysis tasks for future evaluations. The MorphoChallenge 2009 meeting on "Unsupervised Morpheme Analysis" was held on the morning of September 30. The objective of this year's challenge was to design a statistical machine learning algorithm for morpheme discovery. MorphoChallenge is part of the EU Network of Excellence PASCAL Programme.

The CLEF 2008 and 2009 campaigns were organized by TrebleCLEF, a Coordination Action of the Seventh Framework Programme. TrebleCLEF has built on the results achieved by CLEF, supporting the development of expertise in the multidisciplinary research area of multilingual information access and promoting a dissemination action in the relevant application communities. As part of its activities, the project has released a set of Best Practice recommendations in the areas of MLIA System Development and Search Assistance, Test Collection Creation, and Language Processing Technologies. The results of TrebleCLEF can be accessed at www.trebleclef.eu.

This is the first time that the CLEF proceedings are published in two volumes reporting the results of the Text Retrieval Experiments and the Multimedia Experiments, separately. This decision was made necessary by the large participation in CLEF 2009 and our desire to provide an exhaustive overview of all the various activities. This volume reports research and experiments on various types of textual document collections. It is divided into six main sections presenting the results of the following tracks: Multilingual Document Retrieval (Ad-Hoc), Multiple Language Question Answering (QA@CLEF), Multilingual Information Filtering (INFILE@CLEF), Intellectual Property (CLEF-IP) and Log File Analysis (LogCLEF), plus the activities of the Morpho-Challenge program. The companion volume contains the results of the remaining three tracks running on multimedia data: Interactive Cross-Language Retrieval (iCLEF), Cross-Language Image Retrieval (ImageCLEF), and Cross-Language Video Retrieval (VideoCLEF). The table of contents is included in this volume. The papers are mostly extended and revised versions of the initial working notes distributed at the workshop. All papers were subjected to a reviewing procedure. The final volumes were prepared with assistance of the Center for the Evaluation of Language and Communication Technologies (CELCT), Trento, Italy, under the coordination of Danilo Giampiccolo and Pamela Forner. The support of CELCT is gratefully acknowledged. We should also like to thank all the additional reviewers for their careful refereeing.

April 2010

Carol Peters
Giorgio Maria Di Nunzio
Mikko Kurimo
Thomas Mandl
Djamel Mostefa
Anselmo Peñas
Giovanna Roda

Reviewers

The editors express their gratitude to the colleagues listed below for their assistance in reviewing the papers in this volume:

- Eneko Agirre, University of the Basque Country, Spain
- Abolfazl AleAhmad, University of Tehran, Iran
- Iñaki Alegria, University of the Basque Country, Spain
- Giambattista Amati, Fondazione Ugo Bordoni, Italy
- Ebru Arisoy, Bogazici University, Turkey
- Victoria Arranz, ELDA, France
- Sören Auer, University of Leipzig, Germany
- Helmut Berger, Matrixware, Austria
- Delphine Bernhard, LIMSI-CNRS, France
- Romaric Besançon, CEA-LIST, France
- Alessio Bosca, Celi s.r.l., Italy
- Gosse Bouma, Rijksuniversiteit Groningen, The Netherlands
- Burcu Can, University of York, UK
- Stéphane Chaudiron, Université de Lille 3, France
- Tolga Ciloglu, Middle East Technical University, Turkey
- Cagri Coltekin, University of Groningen, The Netherlands
- Luis Fernando Costa, SINTEF ICT, Portugal
- Michael Dittenbach, Matrixware, Austria
- Nicola Ferro, University of Padua, Italy
- Corina Forâscu, Al.I. Cuza University of Iasi, Romania
- M. Rami Ghorab, Trinity College, Ireland
- Ingo Glöckner, FernUniversität in Hagen, Germany
- Erik Graf, University of Glasgow, UK
- Harald Hammarström, Chalmers University, Sweden
- Olivier Hamon, ELDA, France
- Allan Hanbury, Information Retrieval Facility, Austria
- Sven Hartrumpf, FernUniversität in Hagen - IICS, Germany
- Jim Jansen, The Pennsylvania State University, USA
- Kalervo Jarvelin, University of Tampere, Finland
- Dietrich Klakow, Saarland University, Germany
- Oskar Kohonen, Aalto University, Finland
- Katrin Lamm, University of Hildesheim, Germany
- Ray Larson, University of California at Berkeley, USA
- Johannes Leveling, Dublin City University, Ireland
- Constantine Lignos, University of Pennsylvania, USA

- Mihai Lupu, Information Retrieval Facility, Austria
- Thomas Mandl, University of Hildesheim, Germany
- Diego Molla, Macquarie University, Australia
- Christian Monson, Oregon Health & Science University, USA
- Nicolas Moreau, ELDA, France
- Michael Oakes, University of Sunderland, UK
- Constantin Orasan, University of Wolverhampton, UK
- Vivien Petras, Humboldt University, Germany
- Florina Piroi, Information Retrieval Facility, Austria
- Horacio Rodríguez, Polytechnic University of Catalonia, Spain
- Paolo Rosso, Polytechnic University of Valencia, Spain
- Erik Tjong Kim Sang, University of Groningen, Netherlands
- Murat Saraclar, Bogazici University, Turkey
- Julia Maria Schulz, University of Hildesheim, Germany
- Sebastian Spiegler, University of Bristol, UK
- John Tait, Information Retrieval Facility, Austria
- Jordi Turmo, Polytechnic University of Catalonia, Spain
- Jose Luis Vicedo, University of Alicante, Spain
- Sami Virpioja, Aalto University, Finland
- Christa Womser-Hacker, University of Hildesheim, Germany
- Alex Yeh, The MITRE Corporation, USA
- Daniel Zeman, Charles University, Czech Republic
- Veronika Zenz, Matrixware, Austria

CLEF 2009 Coordination

CLEF 2000–2009 was coordinated by the Istituto di Scienza e Tecnologie dell' Informazione, Consiglio Nazionale delle Ricerche, Pisa. The following institutions contributed to the organization of the different tracks of the 2009 campaign:

- Adaptive Informatics Research Centre, Helsinki University of Technology, Finland
- Berlin School of Library and Information Science, Humboldt University, Germany
- Business Information Systems, University of Applied Sciences Western Switzerland, Sierre, Switzerland
- CEA LIST, France
- Center for Autonomous Systems, Royal Institute of Technology, Sweden
- Center for Evaluation of Language and Communication Technologies, Italy
- Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
- Computer Science Department, University of the Basque Country, Spain
- Computer Vision and Multimedia Lab, University of Geneva, Switzerland
- Database Research Group, University of Tehran, Iran
- Department of Computer Science & Information Systems, University of Limerick, Ireland
- Department of Information Engineering, University of Padua, Italy
- Department of Information Science, University of Hildesheim, Germany
- Department of Information Studies, University of Sheffield, UK
- Department of Medical Informatics and Clinical Epidemiology, Oregon Health and Science University, USA
- Department of Medical Informatics, Aachen University of Technology, Germany
- Evaluations and Language Resources Distribution Agency Sarl, Paris, France
- Fraunhofer Institute for Digital Media Technology (IDMT), Germany
- GERiiCO, Université de Lille, France
- Idiap Research Institute, Switzerland
- Information Retrieval Facility (IRF), Austria
- Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI), Orsay, France
- Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
- Lenguajes y Sistemas Informáticos, Universidad Nacional de Educación a Distancia, Madrid, Spain
- Linguateca, SINTEF ICT, Norway
- Linguistic Modelling Laboratory, Bulgarian Academy of Sciences, Bulgaria
- Matrixware Information Services, Austria

- Mediamatics, Delft University of Technology, The Netherlands
- Medical Informatics Service, University Hospitals and University of Geneva, Switzerland
- MITRE Corporation, USA
- National Institute of Standards and Technology, Gaithersburg MD, USA
- NLE Lab., Universidad Politènica de Valencia, Spain
- Research Institute for Artificial Intelligence, Romanian Academy, Romania
- Romanian Institute for Computer Science, Romania
- Royal Institute of Technology (KTH), Stockholm, Sweden
- School of Computing, Dublin City University, Ireland
- Swedish Institute of Computer Science, Sweden
- University of Applied Sciences Western Switzerland (HES-SO), Switzerland

CLEF 2009 Steering Committee

- Maristella Agosti, University of Padua, Italy
- Martin Braschler, Zurich University of Applied Sciences, Switzerland
- Amedeo Cappelli, ISTI-CNR and CELCT, Italy
- Hsin-Hsi Chen, National Taiwan University, Taipei, Taiwan
- Khalid Choukri, Evaluations and Language Resources Distribution Agency, Paris, France
- Paul Clough, University of Sheffield, UK
- Thomas Deselaers, ETH, Switzerland
- Giorgio Di Nunzio, University of Padua, Italy
- David A. Evans, Clairvoyance Corporation, USA
- Marcello Federico, Fondazione Bruno Kessler, Trento, Italy
- Nicola Ferro, University of Padua, Italy
- Christian Fluhr, Cadege, France
- Norbert Fuhr, University of Duisburg, Germany
- Frederic C. Gey, U.C. Berkeley, USA
- Julio Gonzalo, LSI-UNED, Madrid, Spain
- Donna Harman, National Institute of Standards and Technology, USA
- Gareth Jones, Dublin City University, Ireland
- Franciska de Jong, University of Twente, The Netherlands
- Noriko Kando, National Institute of Informatics, Tokyo, Japan
- Jussi Karlgren, Swedish Institute of Computer Science, Sweden
- Michael Kluck, German Institute for International and Security Affairs, Berlin, Germany
- Natalia Loukachevitch, Moscow State University, Russia
- Bernardo Magnini, Fondazione Bruno Kessler, Trento, Italy
- Paul McNamee, Johns Hopkins University, USA
- Henning Müller, University of Applies Sciences Western Switzerland, Sierre and University of Geneva, Switzerland
- Douglas W. Oard, University of Maryland, USA
- Anselmo Peñas, LSI-UNED, Madrid, Spain
- Vivien Petras, Humboldt University Berlin, Germany
- Maarten de Rijke, University of Amsterdam, The Netherlands
- Diana Santos, Linguateca, Sintef, Oslo, Norway
- Jacques Savoy, University of Neuchâtel, Switzerland
- Peter Schäuble, Eurospider Information Technologies, Switzerland
- Richard Sutcliffe, University of Limerick, Ireland

- Hans Uszkoreit, German Research Center for Artificial Intelligence, Germany
- Felisa Verdejo, LSI-UNED, Madrid, Spain
- José Luis Vicedo, University of Alicante, Spain
- Ellen Voorhees, National Institute of Standards and Technology, USA
- Christa Womser-Hacker, University of Hildesheim, Germany

Table of Contents – Part I

AdHoc-Persian

AdHoc-Robust

II: Multiple Language Question Answering (QA@CLEF)

ResPubliQA

QAST

GikiCLEF

III: Multilingual Information Filtering (INFILE)

IV: Intellectual Property (CLEF-IP)

V: Logfile Analysis (LogCLEF)

VI: Grid Experiments (GRID@CLEF)

VII: Morphochallenge

Table of Contents – Part II

ImageCLEFPhoto

ImageCLEFwiki

ImageCLEFmed

ImageCLEFmed Annotation

ImageCLEF Annotation and Robot Vision

ImageCLEF Mixed

III: Cross-Language Retrieval in Video Collections (VideoCLEF)

What Happened in CLEF 2009

Carol Peters

Istituto di Scienza e Tecnologie dell'Informazione (ISTI-CNR), Pisa, Italy
carol.peters@isti.cnr.it

Abstract. The organization of the CLEF 2009 evaluation campaign is described and details are provided concerning the tracks, test collections, evaluation infrastructure, and participation. The aim is to provide the reader of these proceedings with a complete picture of the entire campaign, covering both text and multimedia retrieval experiments. In the final section, the main results achieved by CLEF in the first ten years of activity are discussed and plans for the future of CLEF are presented.

1 Introduction

The objective of the Cross Language Evaluation Forum is to promote research in the field of multilingual system development. This is done through the organisation of annual evaluation campaigns in which a series of tracks designed to test different aspects of mono- and cross-language information retrieval (IR) are offered. The intention is to encourage experimentation with all kinds of multilingual information access – from the development of systems for monolingual retrieval operating on many languages to the implementation of complete multilingual multimedia search services. This has been achieved by offering an increasingly complex and varied set of evaluation tasks over the years. The aim is to meet and anticipate the needs of the multidisciplinary research community working in this area and to encourage the development of next generation multilingual IR systems. CLEF is perhaps one of the few platforms where groups working in many different areas (e.g. Information Retrieval, Natural Language Processing, Image Processing, Speech Recognition, Log Analysis, etc.) have a chance to see what others are doing, and discuss and compare ideas. Figure 1 shows the evolution of CLEF in ten years of activity.

This is the first time that the CLEF post-campaign proceedings have been published in two separate volumes. This decision has been made necessary by the large participation in CLEF 2009 and our desire to provide an exhaustive overview of all the various evaluation activities. We have thus distinguished between papers describing systems and functionality for text retrieval and for multimedia retrieval. This volume reports experiments on various types of textual document collections. It is divided into six main sections presenting the results of the following tracks: Multilingual Document Retrieval (Ad-Hoc), Multiple Language Question Answering (QA@CLEF), Multilingual Information Filtering (INFILE@CLEF), Intellectual Property (CLEF-IP) and Log File Analysis (LogCLEF), plus the activities of the MorphoChallenge program. The papers are mostly extended and revised versions of the initial working notes

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 1–12, 2010.

distributed at the workshop. For details on the results of the tracks conducting experiments on multimedia data: Interactive Cross-Language Retrieval (iCLEF), Cross-Language Image Retrieval (ImageCLEF), and Cross-Language Video Retrieval (VideoCLEF), the reader is referred to the companion volume[1].

This Introduction gives a brief overview of entire campaign in order to provide the reader with a complete picture of what happened: Section 2 lists the various tracks and tasks offered in 2009; Sections 3 and 4 describe the participation and the evaluation infrastructure; the final section gives an assessment of the results achieved by CLEF in this first ten years of activity and presents plans for the future.

2 Tracks and Tasks in CLEF 2009

CLEF 2009 offered eight tracks designed to evaluate the performance of systems for:

- multilingual textual document retrieval (Ad Hoc)
- interactive cross-language retrieval (iCLEF)
- multiple language question answering (QA@CLEF)
- cross-language retrieval in image collections (ImageCLEF)
- multilingual information filtering (INFILE@CLEF)
- cross-language video retrieval (VideoCLEF)
- intellectual property (CLEF-IP) – New this year
- log file analysis (LogCLEF) – New this year

CLEF 2000	▪ mono-, bi- & multilingual text doc retrieval (Ad Hoc) ▪ mono- and cross-language information on structured scientific data (Domain-Specific)
CLEF 2001 **New**	▪ interactive cross-language retrieval (iCLEF)
CLEF 2002 **New**	▪ cross-language spoken document retrieval (CL-SR)
CLEF 2003 **New**	▪ multiple language question answering (QA@CLEF) ▪ cross-language retrieval in image collections (ImageCLEF)
CLEF 2005 **New**	▪ multilingual retrieval of Web documents (WebCLEF) ▪ cross-language geographical retrieval (GeoCLEF)
CLEF 2008 **New**	▪ cross-language video retrieval (VideoCLEF) ▪ multilingual information filtering (INFILE@CLEF)
CLEF 2009 **New**	▪ intellectual property (CLEF-IP) ▪ log file analysis (LogCLEF)

Fig. 1. Evolution of CLEF Tracks

[1] Multilingual Information Access Evaluation II: Multimedia Experiments, LNCS Vol. 6242, Springer.

An experimental pilot task was also offered:

- Grid Experiments (Grid@CLEF)

In addition, Morpho Challenge 2009 was organized in collaboration with CLEF as part of the EU Network of Excellence Pascal Challenge Program[2].

Here below we give a brief overview of the various activities.

Multilingual Textual Document Retrieval (Ad Hoc): The aim of this track has been to promote the development of monolingual and cross-language textual document retrieval systems. From 2000 - 2007, the track used collections of European newspaper and news agency documents. In CLEF 2008, the focus of the track was considerably widened: we introduced very different document collections, a non-European target language, and an information retrieval (IR) task designed to attract participation from groups interested in natural language processing (NLP). Ad Hoc 2009 was to a large extent a repetition of the previous year's activities, with the same three tasks: Tel@CLEF, Persian@CLEF, and Robust-WSD. An important objective was to create good reusable test collections for each of them The track was thus structured in three distinct streams. The first task offered monolingual and cross-language search on library catalog records and was organized in collaboration with The European Library (TEL)[3]. The second task resembled the ad hoc retrieval tasks of previous years but this time the target collection was a Persian newspaper corpora. The third task was the robust activity which used word sense disambiguated (WSD) data. The track was coordinated jointly by ISTI-CNR and Padua University, Italy; the University of the Basque Country, Spain; with the collaboration of the Database Research Group, University of Tehran, Iran.

Interactive Cross-Language Retrieval (iCLEF): In iCLEF, cross-language search capabilities have been studied from a user-inclusive perspective. A central research question has been how best to assist users when searching information written in unknown languages, rather than how best an algorithm can find information written in languages different from the query language. Since 2006, iCLEF has based its experiments on Flickr, a large-scale, web-based image database where image annotations constitute a naturally multilingual folksonomy. In an attempt to encourage greater participation in user-orientated experiments, a new task was designed for 2008 and continued in 2009. The main novelty has been to focus experiments on a shared analysis of a large search log, generated by iCLEF participants from a single search interface provided by the iCLEF organizers. The focus has been, therefore, on search log analysis rather than on system design. The idea has been to study the behaviour of users in an (almost) naturalistic search scenario, having a much larger data set than in previous iCLEF campaigns. The track was coordinated by UNED, Madrid, Spain; Sheffield University, UK; Swedish Institute of Computer Science, Sweden.

[2] MorphoChallenge is part of the EU Network of Excellence Pascal: http://www.cis.hut.fi/morphochallenge2009/

[3] See http://www.theeuropeanlibrary.org/

Multilingual Question Answering (QA@CLEF): This track has offered monolingual and cross-language question answering tasks since 2003. QA@CLEF 2009 proposed three exercises: ResPubliQA, QAST and GikiCLEF:

- ResPubliQA: The hypothetical user considered for this exercise is a person close to the law domain interested in making inquiries on European legislation. Given a pool of 500 independent natural language questions, systems must return the passage that answers each question (not the exact answer) from the JRC-Acquis collection of EU parliamentary documentation. Both questions and documents are translated and aligned for a subset of languages. Participating systems could perform the task in Basque, Bulgarian, English, French, German, Italian, Portuguese, Romanian and Spanish.
- QAST: The aim of the third QAST exercise was to evaluate QA technology in a real multilingual speech scenario in which written and oral questions (factual and definitional) in different languages are formulated against a set of manually and automatically transcribed audio recordings related to speech events in those languages. The scenario proposed was the European Parliament sessions in English, Spanish and French.
- GikiCLEF: Following the previous GikiP pilot at GeoCLEF 2008, the task focused on open list questions over Wikipedia that require geographic reasoning, complex information extraction, and cross-lingual processing, for collections in Bulgarian, Dutch, English, German, Italian, Norwegian (both Bokmål and Nynorsk), Portuguese and Romanian or Spanish.

The track was organized by a number of institutions (one for each target language), and jointly coordinated by CELCT, Trento, Italy, and UNED, Madrid, Spain.

Cross-Language Retrieval in Image Collections (ImageCLEF): This track evaluated retrieval from visual collections; both text and visual retrieval techniques were employed. A number of challenging tasks were offered:

- multilingual ad-hoc retrieval from a photo collection concentrating on diversity in the results;
- a photographic annotation task using a simple ontology;
- retrieval from a large scale, heterogeneous collection of Wikipedia images with user-generated textual metadata;
- medical image retrieval (with visual, semantic and mixed topics in several languages);
- medical image annotation form two databases, a database of chest CTs to detect nodules and a database of x-ray images;
- detection of semantic categories from robotic images (non-annotated collection, concepts to be detected).

A large number of organisations have been involved in the complex coordination of these tasks. They include: Sheffield University, UK; University of Applied Sciences Western Switzerland; Oregon Health and Science University, USA; University of Geneva, Switzerland; CWI, The Netherlands; IDIAP, Switzerland; University of Geneva, Switzerland; Fraunhofer Gesellschaft, Germany; Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands.

Multilingual Information Filtering (INFILE@CLEF): INFILE (INformation, FILtering & Evaluation) was a cross-language adaptive filtering evaluation track sponsored by the French National Research Agency. INFILE has extended the last filtering track of TREC 2002 in a multilingual context. It used a corpus of 100,000 Agence France Press comparable newswires for Arabic, English and French; and evaluation was performed using an automatic querying of test systems with a simulated user feedback. Each system can use the feedback at any time to increase performance. The track was coordinated by the Evaluation and Language resources Distribution Agency (ELDA), France; University of Lille, France; and CEA LIST, France.

Cross-Language Video Retrieval (VideoCLEF): VideoCLEF 2009 was dedicated to developing and evaluating tasks involving access to video content in a multilingual environment. Participants were provided with a corpus of video data (Dutch-language television, predominantly documentaries) accompanied by speech recognition transcripts. In 2009, there were three tasks: "Subject Classification", which involved automatically tagging videos with subject labels; "Affect", which involved classifying videos according to characteristics beyond their semantic content; ``Finding Related Resources Across Languages", which involved linking video to material on the same subject in a different language. The track was jointly coordinated by Delft University of Technology, The Netherlands, and Dublin City University, Ireland.

Intellectual Property (CLEF-IP): This was the first year for the CLEF-IP track. The purpose of the track was twofold: to encourage and facilitate research in the area of patent retrieval by providing a large clean data set for experimentation; to create a large test collection of patents in three main European languages for the evaluation of cross-language information access. The track focused on the task of prior art search. A large test collection for evaluation purposes was created by exploiting patent citations. The collection consists of a corpus of 1,9 million patent documents and 10,000 topics with an average of 6 relevance assessments per topic.

Log File Analysis (LogCLEF): LogCLEF was an evaluation initiative for the analysis of queries and other logged activities as expression of user behaviour. The goal was the analysis and classification of queries in order to understand search behaviour in multilingual contexts and ultimately to improve search systems. The track used log data from the files of The European Library.

Grid Experiments (Grid@CLEF): This experimental pilot has been planned as a long term activity with the aim of: looking at differences across a wide set of languages; identifying best practices for each language; helping other countries to develop their expertise in the IR field and create IR groups. Participants had to conduct experiments according to the CIRCO (Coordinated Information Retrieval Components Orchestration) protocol, an XML-based framework which allows for a distributed, loosely-coupled, and asynchronous experimental evaluation of Information Retrieval (IR) systems. The track was coordinated jointly by University of Padua, Italy, and the National Institute of Standards and Technology, USA.

Unsupervised Morpheme Analysis (Morpho Challenge): Morpheme analysis is particularly useful in speech recognition, information retrieval and machine translation for morphologically rich languages where the amount of different word forms is very large. In Morpho Challenge 2009 unsupervised algorithms that provide morpheme

analyses for words in different languages were evaluated in various practical applications. The evaluations consisted of: 1) a comparison to grammatical morphemes, 2) using morphemes instead of words in information retrieval tasks, and 3) combining morpheme and word based systems in statistical machine translation tasks. The evaluation languages in 2009 were: Finnish, Turkish, German, English and Arabic. The track was coordinated by Helsinki University of Technology and Cambridge University Engineering Department.

Details on the technical infrastructure and the organisation of all these tracks can be found in the track overview reports in this volume, collocated at the beginning of the relevant sections.

3 Test Collections

The CLEF test collections are made up of documents, topics and relevance assessments. The topics are created to simulate particular information needs from which the systems derive the queries to search the document collections. System performance is evaluated by judging the results retrieved in response to a topic with respect to their relevance, and computing the relevant measures, depending on the methodology adopted by the track. The document sets that have been used to build the test collections in CLEF 2009 included:

- A subset of the CLEF multilingual corpus of news documents in 14 European languages (Ad Hoc WSD-Robust task, MorphoChallenge)
- Hamshahri Persian newspaper corpus (Ad Hoc Persian task)
- Library catalog records in English, French, German plus log files provided by The European Library (Ad Hoc TEL task and LogCLEF)
- Log files from the Tumba search engine: http://www.tumba.pt/ (LogCLEF)
- Flickr web-based image database (iCLEF)
- ResPubliQA document collection, a subset of the JRC Acquis corpus of European legislation (QAatCLEF: ResPubliQA)
- Transcripts of European parliamentary sessions in English and Spanish, and French news broadcasts (QAatCLEF: QAST)
- BELGAPICTURE image collection (ImageCLEFPhoto)
- A collection of Wikipedia images and their user-generated textual metadata (ImageCLEFwiki)
- Articles and images from the Radiology and Radiography journals of the RSNA (Radiological Society of North America) (ImageCLEFmed); IRMA collection for medical image annotation (ImageCLEFmedAnnotation); a collection from the Lung Image Database Consortium (LIDC) (ImageCLEFmedAnnotation)
- A collection of FlickR images (ImageCLEFanno)
- A collection of robotics images created from KTH, Sweden (ImageCLEFrobot Vision)
- Dutch and English documentary television programs (VideoCLEF)
- Agence France Press (AFP) comparable newswire stories in Arabic, French and English (INFILE)

- Patent documents in English, French and German from the European Patent Office (CLEF-IP)
- Acknowledgements of the valuable contribution of the data providers is given at the end of this paper.

4 CLEF and TrebleCLEF

CLEF is organized mainly through the voluntary efforts of many different institutions and research groups. However, the central coordination has always received some support from the EU IST programme under the unit for Digital Libraries and Technology Enhanced Learning, mainly within the framework of the DELOS Network of Excellence. CLEF 2008 and 2009 were organized under the auspices of TrebleCLEF, a Coordination Action of the Seventh Framework Programme.

TrebleCLEF has built on the results achieved by CLEF, supporting the development of expertise in the multidisciplinary research area of multilingual information access and promoting dissemination actions in the relevant application communities. The aim has been to:

- Provide applications that need multilingual search solutions with the possibility to identify the technology which is most appropriate
- Assist technology providers to develop competitive multilingual search solutions.

In 2009, the TrebleCLEF activities included the organization of a Summer School on Multilingual Information Access (MLIA) and a MLIA Technology Transfer Day, and the publication of three Best Practices studies:

- Best Practices in Language Resources for Multilingual Information Access
- Best Practices in System and User-oriented Multilingual Information Access
- Best Practices for Test Collection Creation, Evaluation Methodologies and Language Processing Technologies

Information on the activities of TrebleCLEF can be found on the project website[4].

5 Technical Infrastructure

TrebleCLEF has supported a data curation approach within CLEF as an extension to the traditional methodology in order to better manage, preserve, interpret and enrich the scientific data produced, and to effectively promote the transfer of knowledge. The current approach to experimental evaluation is mainly focused on creating comparable experiments and evaluating their performance whereas researchers would also greatly benefit from an integrated vision of the scientific data produced, together with analyses and interpretations, and from the possibility of keeping, re-using, and enriching them with further information. The way in which experimental results are managed, made accessible, exchanged, visualized, interpreted, enriched and referenced is an integral part of the process of knowledge transfer and sharing towards relevant application communities.

[4] http://www.trebleclef.eu/

The University of Padua has thus developed DIRECT: Distributed Information Retrieval Evaluation Campaign Tool[5], a digital library system for managing the scientific data and information resources produced during an evaluation campaign. A preliminary version of DIRECT was introduced into CLEF in 2005 and subsequently tested and developed in the CLEF 2006 and 2007 campaigns. It has been further developed under TrebleCLEF. In 2009, DIRECT managed the technical infrastructure for several of the CLEF tracks and tasks: Ad Hoc, ImageCLEFphoto, GridCLEF, managing:

- the track set-up, harvesting of documents, management of the registration of participants to tracks;
- the submission of experiments, collection of metadata about experiments, and their validation;
- the creation of document pools and the management of relevance assessment;
- the provision of common statistical analysis tools for both organizers and participants in order to allow the comparison of the experiments;
- the provision of common tools for summarizing, producing reports and graphs on the measured performances and conducted analyses.

6 Participation

Researchers from 117 different academic and industrial institutions submitted runs in CLEF 2009: 81 from Europe, 18 from N.America; 16 from Asia, 1 from S.America and 1 from Africa. Figure 2 shows the trend in participation over the years and Figure 3

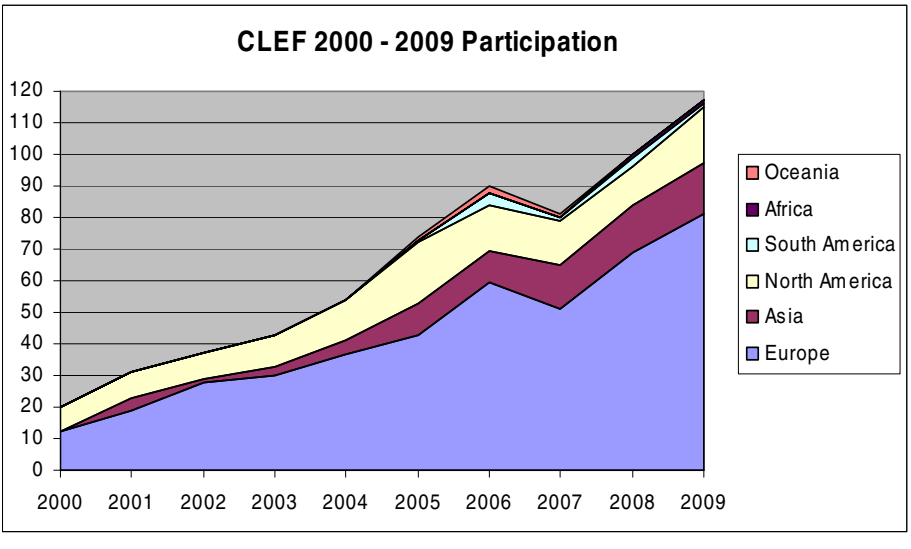


Fig. 2. CLEF 2000 – 2009: Participation

[5] http//direct.dei.unipd.it/

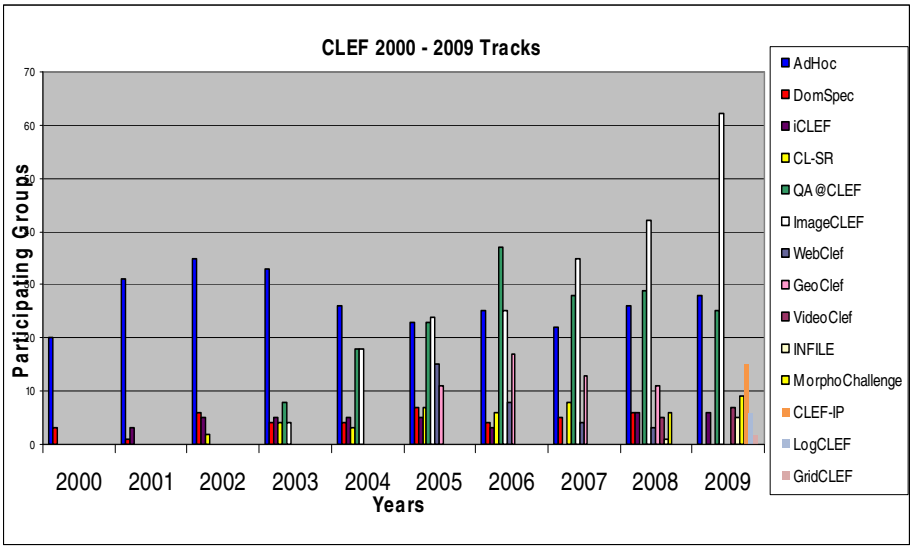


Fig. 3. CLEF 2000 – 2009: Participation per Track

shows the shift in focus as new tracks have been added. As can be seen, the number of groups participating in the Ad Hoc, iCLEF, QA and VideoCLEF tracks is almost the same as last year, there has been a rise of interest in INFILE and participation in the two new tracks (LogCLEF and CLEF-IP) is encouraging.

The most popular track is without doubt ImageCLEF which, with a notable increase from the previous year, tended to dominate the scene in 2009. This gives some cause for reflection as ImageCLEF is the track least concerned with multilinguality. A list of groups and indications of the tracks in which they participated can be found in the CLEF2009 Working Notes on the CLEF website.

7 The Future of CLEF

The main goal of CLEF in this first ten years of activity has been to sustain the growth of excellence in language processing and multilingual information access (MLIA) across language boundaries. A strong motivation has been the desire to promote the study and utilisation of languages other than English on the Internet. In this period, the CLEF activities have produced the following significant results:

- Creation of a very active multidisciplinary international research community, with strong interactions with the other main international initiatives for the evaluation of IR systems: TREC[6], NTCIR[7], and now FIRE[8];

[6] Text REtrieval Conferences, `http://trec.nist.gov/`
[7] NTCIR (NII Test Collection for IR Systems) Project, `http://research.nii.ac.jp/ntcir/`
[8] Forum for Information Retrieval Evaluation, `http://www.isical.ac.in/~clia/`

- Investigation of core issues in MLIA which enable effective transfer over language boundaries, including the development of multiple language processing tools (e.g. stemmers, word decompounders, part-of-speech taggers); creation of linguistic resources (e.g. multilingual dictionaries and corpora); implementation of appropriate cross-language retrieval models and algorithms for different tasks and languages;
- Creation of important reusable test collections and resources in diverse media for a large number of European languages, representative of the major European language typologies;
- Significant and quantifiable improvements in the performance of MLIA systems.

CLEF 2009 has represented an important milestone for the MLIA community. After ten years of activity focused on stimulating the development MLIA systems and functionality through the organisation of increasingly complex evaluation tasks and presenting the results at an annual workshop, we have decided to widen the format. CLEF 2010 will thus take the form of an independent Conference soliciting the submission of papers that propose new retrieval tasks, new evaluation tools, new measures, and new types of operational evaluation, organised in conjunction with a set of Evaluation Labs, which will continue the CLEF tradition of community-based evaluation and discussion on evaluation issues. Two different forms of labs are offered: "campaign-style" labs running evaluation tasks and experiments during the nine month period preceding the conference, and "workshop-style" labs exploring issues of information access evaluation and related fields.

The Conference will be held in Padua, Italy, September 2010, as a four day event: The first two days will consist of plenary sessions in which keynote speeches and peer-reviewed papers will be presented. The goals will be to explore current needs and practices for information access and discuss new directions for future activities in the European multilingual /multimodal IR system evaluation context. In Days 3 and 4, the results of the Labs will be presented in full and half-day workshops. Information on CLEF 2010 is available online[9].

Acknowledgements

It would be impossible to run the CLEF evaluation initiative and organize the annual workshops without considerable assistance from many groups. CLEF is organized on a distributed basis, with different research groups being responsible for the running of the various tracks. My gratitude goes to all those who have been involved in the coordination of the 2009 campaigns. A list of the main institutions involved is given at the beginning of this volume. Here below, let me thank just some of the people responsible for the coordination of the different tracks. My apologies to all those I have not managed to mention:

- Abolfazl AleAhmad, Hadi Amiri, Eneko Agirre, Giorgio Di Nunzio, Nicola Ferro, Nicolas Moreau, Arantxa Otegi and Vivien Petras for the Ad Hoc Track
- Paul Clough, Julio Gonzalo and Jussi Karlgren for iCLEF

[9] http://clef2010.org/

- Iñaki Alegria, Davide Buscaldi, Luís Miguel Cabral, Pere R. Comas, Corina Forascu, Pamela Forner, Olivier Galibert, Danilo Giampiccolo, Nicolas Moreau, Djamel Mostefa, Petya Osenova, Anselmo Peñas, Álvaro Rodrigo, Sophie Rosset, Paolo Rosso, Diana Santos, Richard Sutcliff and Jordi Turmo for QA@CLEF
- Brian Bakke, Steven Bedrick, Barbara Caputo, Paul Clough, Peter Dunker, Thomas Deselaers, Thomas Deserno, Ivan Eggel, Mark Oliver Güld, William Hersh, Patric Jensfelt, Charles E. Kahn Jr., Jana Kludas, Jayashree Kalpathy–Cramer, Henning Müller, Stefanie Nowak, Monica Lestari Paramita, Andrzej Pronobis, Saïd Radhouani, Mark Sanderson, Tatiana Tommasi, Theodora Tsikrika and Petra Welter for ImageCLEF
- Romaric Besançon, Stéphane Chaudiron, Khalid Choukri, Meriama Laïb, Djamel Mostefa and Ismaïl Timimi for INFILE
- Gareth J.F. Jones, Martha Larson and Eamonn Newman for VideoCLEF
- Giovanna Roda, John Tait, Florina Piroi, and Veronika Zenz for CLEF-IP
- Maristella Agosti, Giorgio Di Nunzio, Christine Doran, Inderjeet Mani, Thomas Mandl, Julia Maria Schulz and Alexander Yeh for LogCLEF
- Nicola Ferro and Donna Harman for GridCLEF
- Graeme W. Blackwood, William Byrne Mikko Kurimo, Ville T. Turunen and Sami Virpioja for MorphoChallenge at CLEF
- Marco Duissin, Giorgio Di Nunzio and Nicola Ferro for developing and managing the DIRECT infrastructure.

I also thank all those colleagues who have helped us by preparing topic sets in different languages and the members of the CLEF Steering Committee who have assisted me with their advice and suggestions throughout this campaign.

Furthermore, I gratefully acknowledge the support of all the data providers and copyright holders, and in particular:

- The Los Angeles Times, for the American-English newspaper collection.
- SMG Newspapers (The Herald) for the British-English newspaper collection.
- Le Monde S.A. and ELDA: Evaluations and Language resources Distribution Agency, for the French newspaper collection.
- Frankfurter Rundschau, Druck und Verlagshaus Frankfurt am Main; Der Spiegel, Spiegel Verlag, Hamburg, for the German newspaper collections.
- Hypersystems Srl, Torino and La Stampa, for the Italian newspaper data.
- Agencia EFE S.A. for the Spanish news agency data.
- NRC Handelsblad, Algemeen Dagblad and PCM Landelijke dagbladen/Het Parool for the Dutch newspaper data.
- Aamulehti Oyj and Sanoma Osakeyhtiö for the Finnish newspaper data.
- Russika-Izvestia for the Russian newspaper data.
- Hamshahri newspaper and DBRG, Univ. Tehran, for the Persian newspaper data.
- Público, Portugal, and Linguateca for the Portuguese (PT) newspaper collection.
- Folha, Brazil, and Linguateca for the Portuguese (BR) newspaper collection.
- Tidningarnas Telegrambyrå (TT) SE-105 12 Stockholm, Sweden for the Swedish newspaper data.
- Schweizerische Depeschenagentur, Switzerland, for the French, German & Italian Swiss news agency data.

- Ringier Kiadoi Rt. (Ringier Publishing Inc.).and the Research Institute for Linguistics, Hungarian Acad. Sci. for the Hungarian newspaper documents.
- Sega AD, Sofia; Standart Nyuz AD, Sofia, Novinar OD, Sofia and the BulTreeBank Project, Linguistic Modelling Laboratory, IPP, Bulgarian Acad. Sci, for the Bulgarian newspaper documents
- Mafra a.s. and Lidové Noviny a.s. for the Czech newspaper data
- Usurbilgo Udala, Basque Country, Spain, for the Egunkaria, Basque newspaper documents
- The European Commission – Joint Research Centre for the JRC Acquis Parallel corpus of European legislation in many languages.
- AFP Agence France Presse for the English, French and Arabic newswire data used in the INFILE track
- The British Library, Bibliothèque Nationale de France and the Austrian National Library for the library catalog records forming part of The European Library (TEL)
- The European Library (TEL) for use of TEL log files
- Tumba! web search engine of the Faculdade de Ciências da Universidade de Lisboa (FCUL), Portugal, for logfile querying
- Aachen University of Technology (RWTH), Germany, for the IRMA annotated medical images.
- Radiological Society of North America for the images of the Radiology and Radiographics journals.
- Lung Image Database Consortium (LIDC) for their database of lung nodules.
- Belga Press Agency, Belgium, for BELGAPICTURE image collection
- LIACS Medialab, Leiden University, The Netherlands & Fraunhofer IDMT, Ilmenau, Germany for the use of the MIRFLICKR 25000 Image collection
- Wikipedia for the use of the Wikipedia image collection.
- ELDA for the use of the ESTER Corpus: Manual and automatic transcripts of French broadcast news
- ELDA for the use of EPPS 2005/2006 ES & EN Corpora: Manual and automatic transcriptions of European Parliament Plenary Sessions in Spanish and English
- Matrixware Information Services GmbH for the use of a collection of patent documents in English, French and German from the European Patent Office
- The Institute of Sound and Vision, The Netherlands, for the English/Dutch videos, the University of Twente for the speech transcriptions, and Dublin City University for the shot segmentation.

Without their contribution, this evaluation activity would be impossible.

CLEF 2009 Ad Hoc Track Overview: TEL and Persian Tasks

Nicola Ferro[1] and Carol Peters[2]

[1] Department of Information Engineering, University of Padua, Italy
ferro@dei.unipd.it
[2] ISTI-CNR, Area di Ricerca, Pisa, Italy
carol.peters@isti.cnr.it

Abstract. The design of the 2009 Ad Hoc track was to a large extent a repetition of the previous year's track, with the same three tasks: Tel@CLEF, Persian@CLEF, and Robust-WSD. In this first of the two track overviews, we describe the objectives and results of the TEL and Persian tasks and provide some statistical analyses.

1 Introduction

From 2000 - 2007, the Ad Hoc track at CLEF exclusively used collections of European newspaper and news agency documents[1]. In 2008 it was decided to change the focus and to introduce document collections in a different genre (bibliographic records from The European Library - TEL[2]) and in a non-European language (Persian), and an IR task that would appeal to the NLP community (robust retrieval on word-sense disambiguated data). The 2009 Ad Hoc track has been to a large extent a repetition of the previous year's track, with the same three tasks: Tel@CLEF, Persian@CLEF, and Robust-WSD. An important objective of this two-year period of activity has been to ensure that for each task a good reusable test collections has been created. In this first of the two Ad Hoc track overviews we describe the organisation and results of the TEL and Persian tasks.

TEL@CLEF: This task offered monolingual and cross-language search on library catalogs. It was organized in collaboration with The European Library and used three collections derived from the catalogs of the British Library, the Bibliothéque Nationale de France and the Austrian National Library. Hardly surprisingly, these collections contained records in many languages in addition to the expected English, French or German. The aim of the task was to identify the most effective retrieval technologies for searching this type of very sparse multilingual data. It presumed a user with a working knowledge of these three languages who wants to find documents that can be useful via one of the three target catalogs.

[1] In these eight years, this track built up test collections for monolingual and cross language system evaluation in 14 European languages.
[2] See http://www.theeuropeanlibrary.org/

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 13–35, 2010.

Persian@CLEF: This activity was coordinated again this year in collaboration with the Database Research Group (DBRG) of Tehran University. We chose Persian as our first non-European language target collection for several reasons: its challenging script (a modified version of the Arabic alphabet with elision of short vowels) written from right to left; its complex morphology (extensive use of suffixes and compounding); its political and cultural importance. A more detailed description of the specific characteristics of the Persian language and the challenges it poses for information retrieval are given in [13]. The task used the Hamshahri corpus of 1996-2002 newspapers as the target collection and was organised as a traditional ad hoc document retrieval task. Monolingual and cross-language (English to Persian) tasks were offered.

In the rest of this paper we present the task setup, the evaluation methodology and the participation in the two tasks (Section 2). We then describe the main features of each task and show the results (Sections 3 and 4). The final section provides a brief summing up. For information on the various approaches and resources used by the groups participating in the two tasks and the issues they focused on, we refer the reader to the papers in the relevant Ad Hoc sections of these Proceedings or in the CLEF 2009 Working Notes[3].

2 Track Setup

As is customary in the CLEF Ad Hoc track, we adopted a corpus-based, automatic scoring method for the assessment of the performance of the participating systems, based on ideas first introduced in the Cranfield experiments in the late 1960s [5]. The tasks offered are studied in order to effectively measure textual document retrieval under specific conditions. The test collections are made up of documents, topics and relevance assessments. The topics consist of a set of statements simulating information needs from which the systems derive the queries to search the document collections. Evaluation of system performance is then done by judging the documents retrieved in response to a topic with respect to their relevance, and computing the recall and precision measures. The pooling methodology is used in order to limit the number of manual relevance assessments that have to be made. As always, the distinguishing feature of CLEF is that it applies this evaluation paradigm in a multilingual setting. This means that the criteria normally adopted to create a test collection, consisting of suitable documents, sample queries and relevance assessments, have been adapted to satisfy the particular requirements of the multilingual context. All language dependent tasks such as topic creation and relevance judgment are performed in a distributed setting by native speakers. Rules are established and a tight central coordination is maintained in order to ensure consistency and coherency of topic and relevance judgment sets over the different collections, languages and tracks.

2.1 The Documents

As mentioned in the Introduction, the two tasks used different sets of documents.

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

The TEL task used three collections:

- British Library (BL); 1,000,100 documents, 1.2 GB;
- Bibliothéque Nationale de France (BNF); 1,000,100 documents, 1.3 GB;
- Austrian National Library (ONB); 869,353 documents, 1.3 GB.

We refer to the three collections (BL, BNF, ONB) as English, French and German because, in each case, this is the main and expected language of the collection. However, as has been mentioned, each of these collections is to some extent multilingual and contains documents (catalog records) in many additional languages.

The TEL data is very different from the newspaper articles and news agency dispatches previously used in the CLEF ad hoc track. The data tends to be very sparse. Many records contain only title, author and subject heading information; other records provide more detail. The title and (if existing) an abstract or description may be in a different language to that understood as the language of the collection. The subject heading information is normally in the main language of the collection. About 66% of the documents in the English and German collection have textual subject headings, while only 37% in the French collection. Dewey Classification (DDC) is not available in the French collection; negligible (<0.3%) in the German collection; but occurs in about half of the English documents (456,408 docs to be exact).

Whereas in the traditional ad hoc task, the user searches directly for a document containing information of interest, here the user tries to identify which publications are of potential interest according to the information provided by the catalog card. When we designed the task, the question the user was presumed to be asking was "Is the publication described by the bibliographic record relevant to my information need?"

The Persian task used the Hamshahri corpus of 1996-2002 newspapers as the target collection. This corpus was made available to CLEF by the Data Base Research Group (DBRG) of the University of Tehran. Hamshahri is one of the most popular daily newspapers in Iran. The Hamshahri corpus consists of 345 MB of news texts for the years 1996 to 2002 (corpus size with tags is 564 MB). This corpus contains more than 160,000 news articles about a variety of subjects and includes nearly 417,000 different words. Hamshahri articles vary between 1KB and 140KB in size[4].

2.2 Topics

Topics in the CLEF ad hoc track are structured statements representing information needs. Each topic typically consists of three parts: a brief "title" statement; a one-sentence "description"; a more complex "narrative" specifying the relevance assessment criteria.

For the TEL task, a common set of 50 topics was prepared in each of the 3 main collection languages (English, French and German) plus this year also in

[4] For more information, see http://ece.ut.ac.ir/dbrg/hamshahri/

Chinese, Italian and Greek in response to specific requests. Only the Title and Description fields were released to the participants. The narrative was prepared to provide information for the assessors on how the topics should be judged but was not released to the participants. The topic sets were prepared on the basis of the contents of the collections.

In ad hoc, when a task uses data collections in more than one language, we consider it important to be able to use versions of the same core topic set to query all collections. This makes it easier to compare results over different collections and also facilitates the preparation of extra topic sets in additional languages. However, it is never easy to find topics that are effective for several different collections and the topic preparation stage requires considerable discussion between the coordinators for each collection in order to identify suitable common candidates. The sparseness of the data makes this particularly difficult for the TEL task and leads to the formulation of topics that are quite broad in scope so that at least some relevant documents could be found in each collection. A result of this strategy is that there tends to be a considerable lack of evenness of distribution in relevant documents. For each topic, the results expected from the separate collections can vary considerably. An example of a CLEF 2009 TEL topic in six languages is given in Figure 1.

For the Persian task, 50 topics were created in Persian by the Data Base Research group of the University of Tehran, and then translated into English. The rule in CLEF when creating topics in additional languages is not to produce literal translations but to attempt to render them as naturally as possible. This was a particularly difficult task when going from Persian to English as cultural differences had to be catered for. An example of a CLEF 2009 Persian topic in English and Farsi is given in Figure 2.

2.3 Relevance Assessment

The number of documents in large test collections such as CLEF makes it impractical to judge every document for relevance. Instead approximate recall values are calculated using pooling techniques. The results submitted by the groups participating in the ad hoc tasks are used to form a pool of documents for each topic and language by collecting the highly ranked documents from selected runs according to a set of predefined criteria. One important limitation when forming the pools is the number of documents to be assessed. Traditionally, the top 100 ranked documents from each of the runs selected are included in the pool; in such a case we say that the pool is of depth 100. This pool is then used for subsequent relevance judgments. After calculating the effectiveness measures, the results are analyzed and run statistics produced and distributed. The stability of pools constructed in this way and their reliability for post-campaign experiments is discussed in [3] with respect to the CLEF 2003 pools.

The main criteria used when constructing the pools in CLEF are:

– favour diversity among approaches adopted by participants, according to the descriptions that they provide of their experiments;

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<topic>
    <identifier>10.2452/711-AH</identifier>

    <title lang="zh">深海生物</title>
    <title lang="en">Deep Sea Creatures</title>
    <title lang="fr">Créatures des fonds océaniques</title>
    <title lang="de">Kreaturen der Tiefsee</title>
    <title lang="el">Πλάσματα στα βάθη των ωκεανών</title>
    <title lang="it">Creature delle profondità oceaniche</title>

    <description lang="zh">
        查找有关世界上任何深海生物的出版物。
    </description>
    <description lang="en">
        Find publications about any kind of life in the depths
        of any of the world's oceans.
    </description>
    <description lang="fr">
        Trouver des ouvrages sur toute forme de vie dans les
        profondeurs des mers et des océans.
    </description>
    <description lang="de">
        Finden Sie Veröffentlichungen über Leben und
        Lebensformen in den Tiefen der Ozeane der Welt.
    </description>
    <description lang="el">
        Αναζήτηση δημοσιεύσεων για κάθε είδος ζωής στα
        βάθη των ωκεανών
    </description>
    <description lang="it">
        Trova pubblicazioni su qualsiasi forma di vita nelle
        profondità degli oceani del mondo.
    </description>
</topic>
```

Fig. 1. Example of TEL topic

- for each task, include at least one experiment from every participant, selected from the experiments indicated by the participants as having highest priority;
- ensure that, for each participant, at least one mandatory title+description experiment is included, even if not indicated as having high priority;
- add manual experiments, when provided;
- for bilingual tasks, ensure that each source topic language is represented.

From our experience in CLEF, using the tools provided by the DIRECT system [1], we find that for newspaper documents, assessors can normally judge from 60 to 100 documents per hour, providing binary judgments: relevant / not relevant. Our estimate for the TEL catalog records is higher as these records are much shorter than the average newspaper article (100 to 120 documents per hour). In both cases, it is clear that human relevance assessment is a time-consuming and

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<topic>
    <identifier>10.2452/641-AH</identifier>

    <title lang="en">Pollution in the Persian Gulf</title>
    <title lang="fa">وضعيت آلودگي درياي خليج فارس</title>

    <description lang="en">
        Find information about pollution in the Persian Gulf and the causes.
    </description>
    <description lang="fa">
        بررسي وضعيت درياي خليج فارس از نظر آلودگي و عوامل آن
    </description>

    <narrative lang="en">
        Find information about conditions of the Persian Gulf with respect to
        pollution; also of interest is information on the causes of pollution
        and comparisons of the level of pollution in this sea against that of
        other seas.
    </narrative>
    <narrative lang="fa">
        يافتن اطلاعاتي در مورد وضعيت آلودگي درياي خليج فارس و بررسي عوامل ايجاد آل
        ودگي در اين دريا و اطلاعاتي مقايسه نظير آن با ساير دريا
    </narrative>
</topic>
```

Fig. 2. Example of Persian topic

resource expensive task . This limitation impacts strongly on the application of the criteria above - and implies that we are obliged to be flexible in the number of documents judged per selected run for individual pools.

This year, in order to create pools of more-or-less equivalent size, the depth selected for the TEL English, French, and German pools was 60[5]. For each collection, we included in the pool two monolingual and one bilingual experiment for every participant, plus any documents assessed as relevant during topic creation. As we only had a relatively small number of runs submitted for Persian, we were able to include documents from all experiments, and the pool was created with a depth of 80.

These pool depths were the same as those created in the previous year. Given the resources available, it was not possible to manually assess more documents. For the CLEF 2008 ad hoc test collections, Stephen Tomlinson reported some sampling experiments aimed at estimating the judging coverage [9]. He found that this tended to be lower than the estimates he produced for the CLEF 2007 ad hoc collections. With respect to the TEL collections, he estimated that at best 50% to 70% of the relevant documents were included in the pools - and that most of the unjudged relevant documents were for the 10 or more queries that had the most known answers. Tomlinson has repeated these experiments for the 2009 TEL and Persian data [10]. Although for two of the four languages concerned (German and Persian), his findings were similar to last year's estimates, for the

[5] Tests made on NTCIR pools in previous years have suggested that a depth of 60 is normally adequate to create stable pools, presuming that a sufficient number of runs from different systems have been included.

other two languages (English and French) this year's estimates are substantially lower.

With respect to Tomlinson's analyses, the different nature of the TEL document collections with respect to the "traditional" newspaper collections used in CLEF up to 2007 must be remembered. Although the TEL documents tend to be very sparse they can vary considerably, ranging from very short catalog records to quite long records with full abstracts of the related publications. Moreover, as already stated, each collection is inherently multilingual, and this means that for any topic there may be relevant documents in several languages. This complicates pool construction and the assessment activity because, for example, for the English collection you might have relevant documents for a given topic also in Czech and Hungarian. On the other hand this also makes the task more challenging for the systems: if they focus only on the main language of a collection they are going to target about the 60%-70% of the documents in the collections, leaving out a 30%-40% of potentially relevant documents. This, in turn, will impact the pools created from those systems. If we are to continue to use the pooling technique for this type of collection, we need to do some more exhaustive manual searches in order to boost the pools with respect to relevant documents. We also need to consider more carefully other techniques for relevance assessment in the future such as, for example, the method suggested by Sanderson and Joho [8] or Mechanical Turk [2].

The problem noted with the Persian pool may well be a consequence of the poor participation in this task in 2009. In order to create a stable test collection, you need a good number of runs from systems using different IR models and techniques.

Table 1 reports summary information on the 2009 ad hoc pools used to calculate the results for the main monolingual and bilingual experiments. For each pool, we show the number of topics, the number of runs submitted, the number of runs included in the pool, the number of documents in the pool (relevant and non-relevant), and the number of assessors.

The box plot of Figure 3 compares the distributions of the relevant documents across the topics of each pool for the different ad hoc pools; the boxes are ordered by decreasing mean number of relevant documents per topic.

Figure 4 compares, for each topic, the number of relevant documents in each of the CLEF 2009 TEL collections. We see that French and German distributions appear similar and are slightly asymmetric towards topics with a greater number of relevant documents while the English distribution is slightly asymmetric towards topics with a lower number of relevant documents. All the distributions show some upper outliers, i.e. topics with a greater number of relevant document with respect to the behaviour of the other topics in the distribution. These outliers are probably due to the fact that CLEF topics have to be able to retrieve relevant documents in all the collections; therefore, they may be considerably broader in one collection compared with others, depending on the contents of the separate datasets. As can be seen in the figure, there are very few cases of topics with almost the same number of relevant documents in all the collections.

Table 1. Summary information on CLEF 2009 pools

TEL English Pool (**DOI** 10.2454/AH-TEL-ENGLISH-CLEF2009)	
Pool size	26,190 pooled documents – 23,663 not relevant documents – 2,527 relevant documents 50 topics
Pooled Experiments	31 out of 89 submitted experiments – monolingual: 22 out of 43 submitted experiments – bilingual: 9 out of 46 submitted experiments
Assessors	4 assessors
TEL French Pool (**DOI** 10.2454/AH-TEL-FRENCH-CLEF2009)	
Pool size	21,971 pooled documents – 20,118 not relevant documents – 1,853 relevant documents 50 topics
Pooled Experiments	21 out of 61 submitted experiments – monolingual: 16 out of 35 submitted experiments – bilingual: 5 out of 26 submitted experiments
Assessors	1 assessor
TEL German Pool (**DOI** 10.2454/AH-TEL-GERMAN-CLEF2009)	
Pool size	25,541 pooled documents – 23,882 not relevant documents – 1,559 relevant documents 50 topics
Pooled Experiments	21 out of 61 submitted experiments – monolingual: 16 out of 35 submitted experiments – bilingual: 5 out of 26 submitted experiments
Assessors	2 assessors
Persian Pool (**DOI** 10.2454/AH-PERSIAN-CLEF2009)	
Pool size	23,536 pooled documents – 19,072 not relevant documents – 4,464 relevant documents 50 topics
Pooled Experiments	20 out of 20 submitted experiments – monolingual: 17 out of 17 submitted experiments – bilingual: 3 out of 3 submitted experiments
Assessors	23 assessors

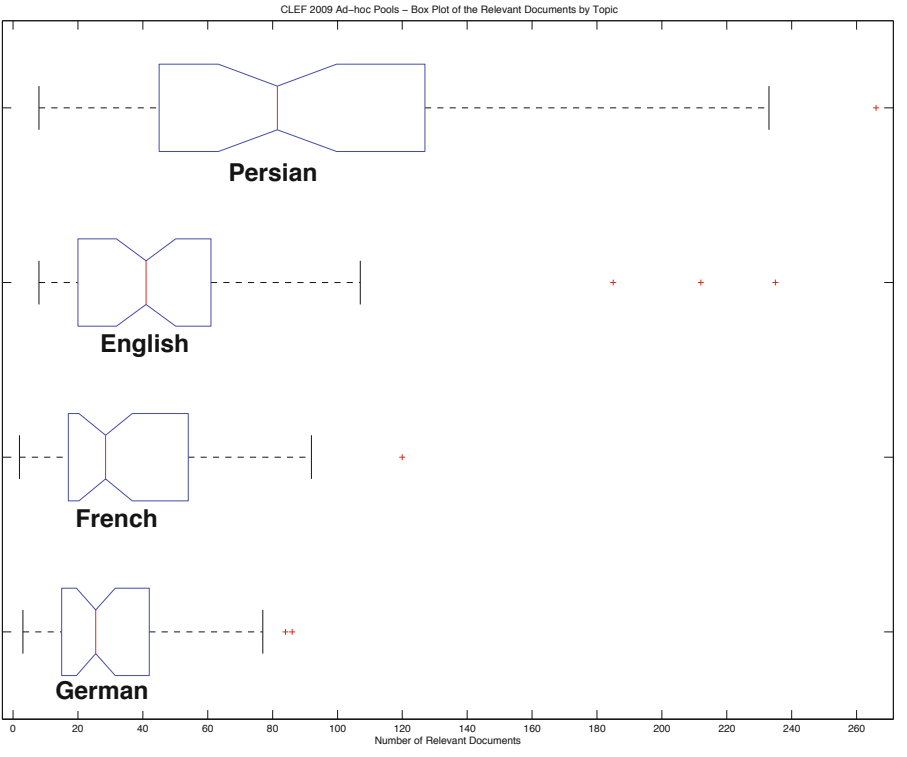


Fig. 3. Distribution of the relevant documents across the ad-hoc pools

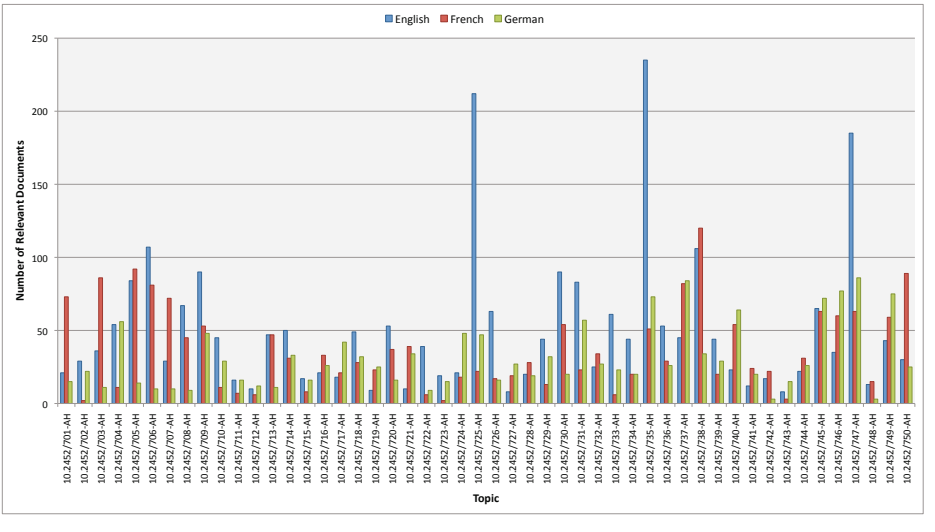


Fig. 4. Comparison topic-by-topic over the number of relevant documents for the TEL pool

The creation of topics with an even distribution of relevant documents across collections in different languages is very difficult and, in fact, not necessary. The goal is to ensure that each test collection is stable and that each topic finds an acceptable number of relevant docs for each collection (but the acceptable number can vary considerably - from few to very many for the same topic).

For the TEL documents, we judged for relevance only those documents that are written totally or partially in English, French and German, e.g. a catalog record written entirely in Hungarian was counted as not relevant as it was of no use to our hypothetical user; however, a catalog record with perhaps the title and a brief description in Hungarian, but with subject descriptors in French, German or English was judged for relevance as it could be potentially useful. Our assessors had no additional knowledge of the documents referred to by the catalog records (or surrogates) contained in the collection. They judged for relevance on the information contained in the records made available to the systems. This was a non trivial task due to the lack of information present in the documents. During the relevance assessment activity there was much consultation between the assessors for the three TEL collections in order to ensure that the same assessment criteria were adopted by everyone.

As shown in the box plot of Figure 3, the Persian distribution presents a greater number of relevant documents per topic with respect to the other distributions and is slightly asymmetric towards topics with a number of relevant documents. In addition, as can be seen from Table 1, it has been possible to sample all the experiments submitted for the Persian tasks. This means that there were fewer unique documents per run and this fact, together with the greater number of relevant documents per topic suggests either that all the systems were using similar approaches and retrieval algorithms or that the systems found the Persian topics quite easy.

The relevance assessment for the Persian results was done by the DBRG group in Tehran. Again, assessment was performed on a binary basis and the standard CLEF assessment rules were applied.

2.4 Result Calculation

Evaluation campaigns such as TREC and CLEF are based on the belief that the effectiveness of *Information Retrieval Systems (IRSs)* can be objectively evaluated by an analysis of a representative set of sample search results. For this, effectiveness measures are calculated based on the results submitted by the participants and the relevance assessments. Popular measures usually adopted for exercises of this type are Recall and Precision. Details on how they are calculated for CLEF are given in [4].

The individual results for all official Ad-hoc TEL and Persian experiments in CLEF 2009 are given in the Appendices of the CLEF 2009 Working Notes [6,7]. You can also access online all the results, topics, experiment, and relevance judgements by logging into `http://direct.dei.unipd.it/`[6]

[6] If you need an account to access the system, please send an e-mail to `direct@dei.unipd.it`.

Table 2. CLEF 2009 Ad hoc Participants

Ad hoc TEL Participants		
Participant	**Institution**	**Country**
aeb	Athens Univ. Economics & Business	Greece
celi	CELI Research srl	Italy
chemnitz	Chemnitz University of Technology	Germany
cheshire	U.C.Berkeley	United States
cuza	Alexandru Ioan Cuza University	Romania
hit	HIT2Lab, Heilongjiang Inst. Tech.	China
inesc	Tech. Univ. Lisbon	Portugal
karlsruhe	Univ. Karlsruhe	Germany
opentext	OpenText Corp.	Canada
qazviniau	Islamic Azaz Univ. Qazvin	Iran
trinity	Trinity Coll. Dublin	Ireland
trinity-dcu	Trinity Coll. & DCU	Ireland
weimar	Bauhaus Univ. Weimar	Germany
Ad hoc Persian Participants		
Participant	**Institution**	**Country**
jhu-apl	Johns Hopkins Univ.	USA
opentext	OpenText Corp.	Canada
qazviniau	Islamic Azaz Univ. Qazvin	Iran
unine	U.Neuchatel-Informatics	Switzerland

2.5 Participants and Experiments

As shown in Table 2, a total of 13 groups from 10 countries submitted official results for the TEL task, while just four groups participated in the Persian task.

A total of 231 runs were submitted with an average number of submitted runs per participant of 13.5 runs/participant.

Participants were required to submit at least one title+description ("TD") run per task in order to increase comparability between experiments. The large majority of runs (216 out of 231, 93.50%) used this combination of topic fields, 2 (0.80%) used all fields[7], and 13 (5.6%) used the title field. All the experiments were conducted using automatic query construction. A breakdown into the separate tasks and topic languages is shown in Table 3.

Seven different topic languages were used in the ad hoc experiments. As always, the most popular language for queries was English, with German second. However, it must be noted that English topics were provided for both the TEL and the Persian tasks. It is thus hardly surprising that English is the most used language in which to formulate queries.

3 TEL@CLEF

The objective of this activity was to search and retrieve relevant items from collections of library catalog cards. The underlying aim was to identify the most

[7] The narrative field was only offered for the Persian task.

Table 3. Number of experiments by task and topic language and number of participants per task

Task	Chinese	English	Farsi	French	German	Greek	Italian	Runs	Part.
TEL Mono English	–	46	–	–	–	–	–	**46**	12
TEL Mono French	–	–	–	35	–	–	–	**35**	9
TEL Mono German	–	–	–	–	35	–	–	**35**	9
TEL Bili English	3	0	0	15	19	5	1	**43**	10
TEL Bili French	0	12	0	0	12	0	2	**26**	6
TEL Bili German	1	12	0	12	0	0	1	**26**	6
Mono Persian	–	–	17	–	–	–	–	**17**	4
Bili Persian	–	3	–	–	–	–	–	**3**	1
Total	**4**	**73**	**17**	**62**	**66**	**5**	**4**	**231**	–

effective retrieval technologies for searching this type of very sparse multilingual data.

3.1 Tasks

Two subtasks were offered which we called Monolingual and Bilingual. In both tasks, the aim was to retrieve documents relevant to the query. By monolingual we mean that the query is in the same language as the main language of the collection. By bilingual we mean that the query is in a different language to the main language of the collection. For example, in an EN → FR run, relevant documents (bibliographic records) could be any document in the BNF collection (referred to as the French collection), in whatever language they are written. The same is true for a monolingual FR → FR run - relevant documents from the BNF collection could actually also be in English or German, not just French.

Ten of the thirteen participating groups attempted a cross-language task; the most popular being with the British Library as the target collection. Six groups submitted experiments for all six possible official cross-language combinations. In addition, we had runs submitted to the BL target collection with queries in Greek, Chinese and Italian.

3.2 Results

Monolingual Results

Table 4 shows the top five groups for each target collection, ordered by mean average precision. The table reports: the short name of the participating group; the mean average precision achieved by the experiment; the DOI of the experiment; and the performance difference between the first and the last participant. Figures 5, 7, and 9 compare the performances of the top participants of the TEL Monolingual tasks.

Table 4. Best entries for the monolingual TEL tasks

Track	Rank	Participant	Experiment DOI	MAP
English	1st	inesc	10.2415/AH-TEL-MONO-EN-CLEF2009.INESC.RUN11	40.84%
	2nd	chemnitz	10.2415/AH-TEL-MONO-EN-CLEF2009.CHEMNITZ.CUT_11_MONO_MERGED_EN_9_10	40.71%
	3rd	trinity	10.2415/AH-TEL-MONO-EN-CLEF2009.TRINITY.TCDENRUN2	40.35%
	4th	hit	10.2415/AH-TEL-MONO-EN-CLEF2009.HIT.MTDD10T40	39.36%
	5th	trinity-dcu	10.2415/AH-TEL-MONO-EN-CLEF2009.TRINITY-DCU.TCDDCUEN3	36.96%
	Difference			10.50%
French	1st	karlsruhe	10.2415/AH-TEL-MONO-FR-CLEF2009.KARLSRUHE.INDEXBL	27.20%
	2nd	chemnitz	10.2415/AH-TEL-MONO-FR-CLEF2009.CHEMNITZ.CUT_19_MONO_MERGED_FR_17_18	25.83%
	3rd	inesc	10.2415/AH-TEL-MONO-FR-CLEF2009.INESC.RUN12	25.11%
	4th	opentext	10.2415/AH-TEL-MONO-FR-CLEF2009.OPENTEXT.OTFR09TDE	24.12%
	5th	celi	10.2415/AH-TEL-MONO-FR-CLEF2009.CELI.CACAO_FRBNF_ML	23.61%
	Difference			15.20%
German	1st	opentext	10.2415/AH-TEL-MONO-DE-CLEF2009.OPENTEXT.OTDE09TDE	28.68%
	2nd	chemnitz	10.2415/AH-TEL-MONO-DE-CLEF2009.CHEMNITZ.CUT_3_MONO_MERGED_DE_1_2	27.89%
	3rd	inesc	10.2415/AH-TEL-MONO-DE-CLEF2009.INESC.RUN12	27.85%
	4th	trinity-dcu	10.2415/AH-TEL-MONO-DE-CLEF2009.TRINITY-DCU.TCDDCUDE3	26.86%
	5th	trinity	10.2415/AH-TEL-MONO-DE-CLEF2009.TRINITY.TCDDERUN1	25.77%
	Difference			11.30%

Bilingual Results

Table 5 shows the top five groups for each target collection, ordered by mean average precision. The table reports: the short name of the participating group; the mean average precision achieved by the experiment; the DOI of the experiment; and the performance difference between the first and the last participant. Figures 6, 8, and 10 compare the performances of the top participants of the TEL Bilingual tasks.

For bilingual retrieval evaluation, a common method is to compare results against monolingual baselines. We have the following results for CLEF 2009:

- X → EN: 99.07% of best monolingual English IR system;
- X → FR: 94.00% of best monolingual French IR system;
- X → DE: 90.06% of best monolingual German IR system.

These figures are very encouraging, especially when compared with the results for last year for the same TEL tasks:

- X → EN: 90.99% of best monolingual English IR system;
- X → FR: 56.63% of best monolingual French IR system;
- X → DE: 53.15% of best monolingual German IR system.

In particular, it can be seen that there is a considerable improvement in performance for French and German.

The monolingual performance figures for all three tasks are quite similar to those of last year but as these are not absolute values, no real conclusion can be drawn from this.

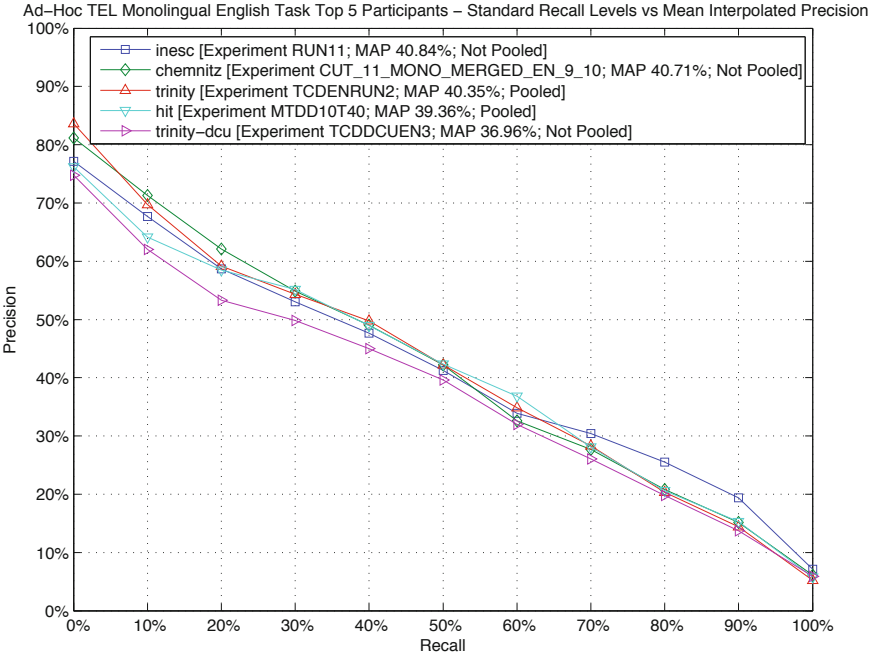


Fig. 5. Monolingual English

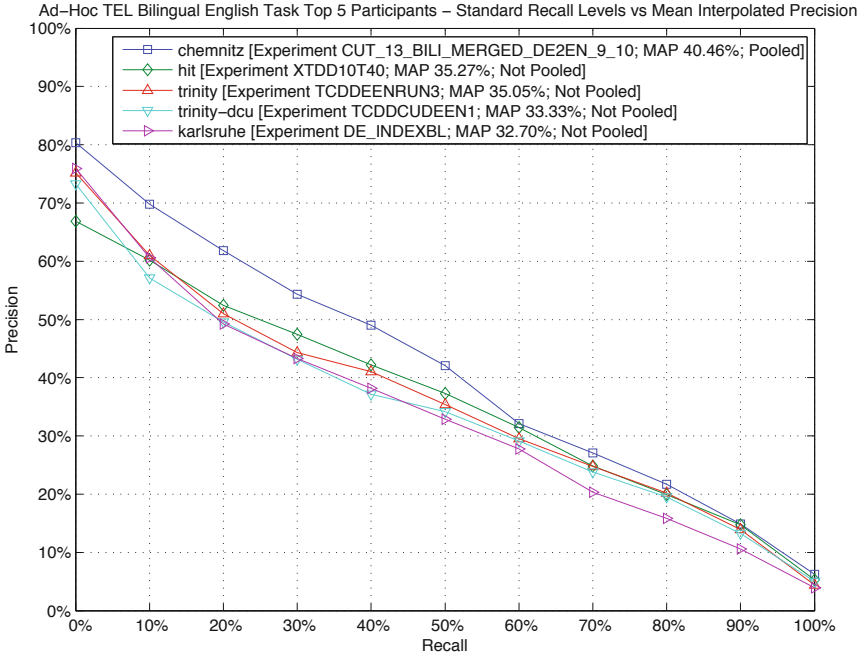


Fig. 6. Bilingual English

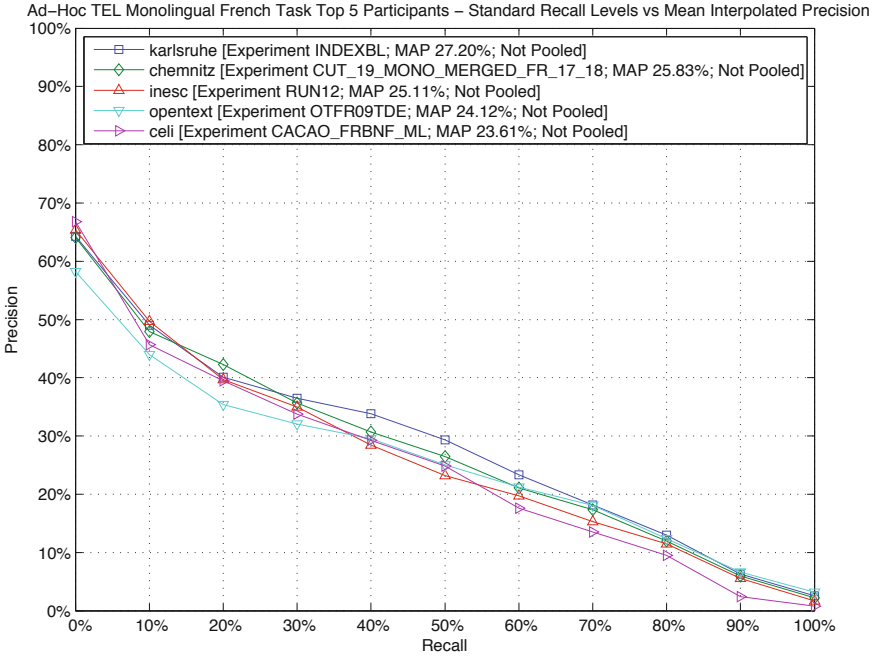


Fig. 7. Monolingual French

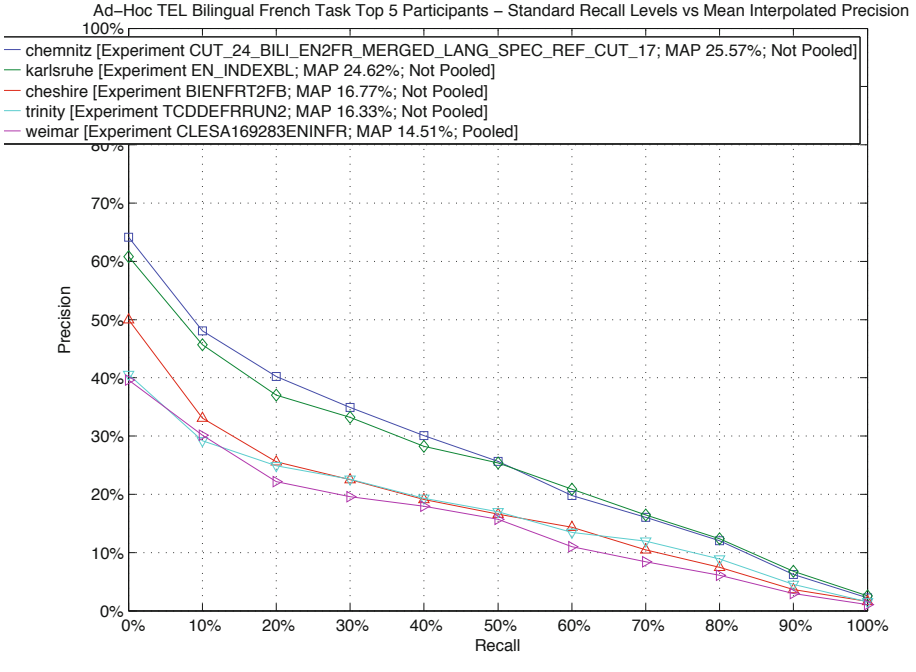


Fig. 8. Bilingual French

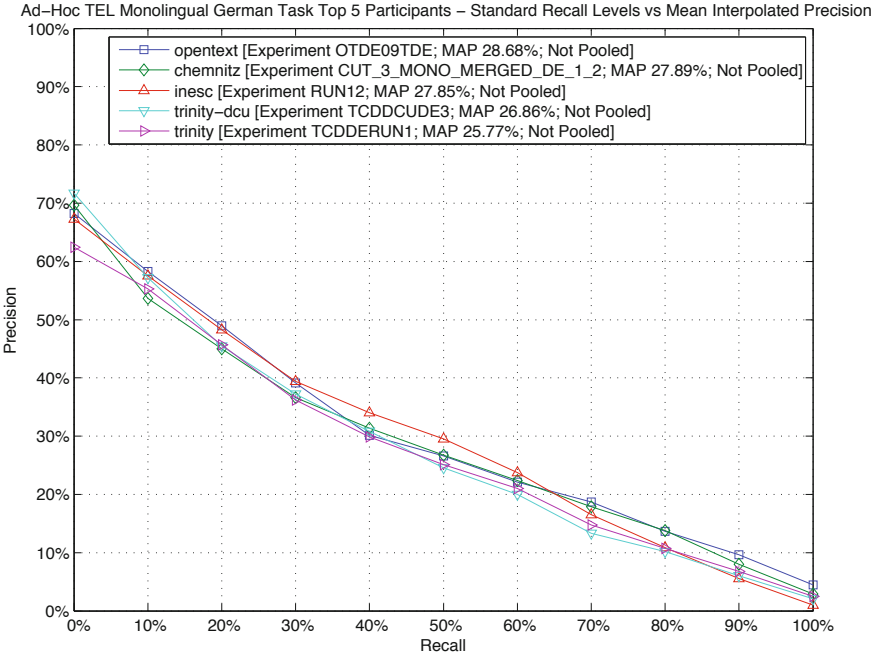


Fig. 9. Monolingual German

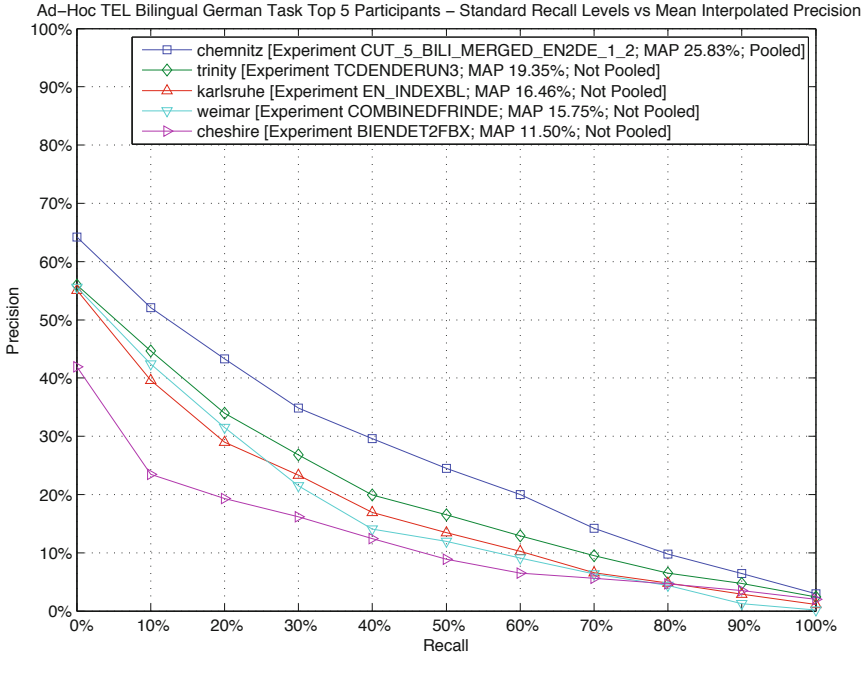


Fig. 10. Bilingual German

Table 5. Best entries for the bilingual TEL tasks

Track	Rank	Participant	Experiment DOI	MAP
English	1st	chemnitz	10.2415/AH-TEL-BILI-X2EN-CLEF2009.CHEMNITZ.CUT_13_BILI_MERGED_DE2EN_9_10	40.46%
	2nd	hit	10.2415/AH-TEL-BILI-X2EN-CLEF2009.HIT.XTDD10T40	35.27%
	3rd	trinity	10.2415/AH-TEL-BILI-X2EN-CLEF2009.TRINITY.TCDDEENRUN3	35.05%
	4th	trinity-dcu	10.2415/AH-TEL-BILI-X2EN-CLEF2009.TRINITY-DCU.TCDDCUDEEN1	33.33%
	5th	karlsrhue	10.2415/AH-TEL-BILI-X2EN-CLEF2009.KARLSRUHE.DE_INDEXBL	32.70%
	Difference			23.73%
French	1st	chemnitz	10.2415/AH-TEL-BILI-X2FR-CLEF2009.CHEMNITZ.CUT_24_BILI_EN2FR_MERGED_LANG_SPEC_REF_CUT_17	25.57%
	2nd	karlsrhue	10.2415/AH-TEL-BILI-X2FR-CLEF2009.KARLSRUHE.EN_INDEXBL	24.62%
	3rd	chesire	10.2415/AH-TEL-BILI-X2FR-CLEF2009.CHESHIRE.BIENFRT2FB	16.77%
	4th	trinity	10.2415/AH-TEL-BILI-X2FR-CLEF2009.TRINITY.TCDDEFRRUN2	16.33%
	5th	weimar	10.2415/AH-TEL-BILI-X2FR-CLEF2009.WEIMAR.CLESA169283ENINFR	14.51%
	Difference			69.67%
German	1st	chemnitz	10.2415/AH-TEL-BILI-X2DE-CLEF2009.CHEMNITZ.CUT_5_BILI_MERGED_EN2DE_1_2	25.83%
	2nd	trinity	10.2415/AH-TEL-BILI-X2DE-CLEF2009.TRINITY.TCDENDERUN3	19.35%
	3rd	karlsrhue	10.2415/AH-TEL-BILI-X2DE-CLEF2009.KARLSRUHE.EN_INDEXBL	16.46%
	4th	weimar	10.2415/AH-TEL-BILI-X2DE-CLEF2009.WEIMAR.COMBINEDFRINDE	15.75%
	5th	chesire	10.2415/AH-TEL-BILI-X2DE-CLEF2009.CHESHIRE.BIENDET2FBX	11.50%
	Difference			124.60%

3.3 Approaches

As stated in the introduction, the TEL task this year is a repetition of the task set last year. A main reason for this was to create a good reusable test collection with a sufficient number of topics; another reason was to see whether the experience gained and reported in the literature last year, and the opportunity to use last year's test collection as training data, would lead to differences in approaches and/or improvements in performance this year. Although we have exactly the same number of participants this year as last year, only five of the thirteen 2009 participants also participated in 2008. These are the groups tagged as Chemnitz, Cheshire, Karlsruhe, INESC and Opentext. The last two of these groups only tackled monolingual tasks. These groups all tend to appear in the top five for the various tasks. In the following we attempt to examine briefly the approaches adopted this year, focusing mainly on the cross-language experiments.

In the TEL task in CLEF 2008, we noted that all the traditional approaches to monolingual and cross language retrieval were attempted by the different groups. Retrieval methods included language models, vector-space and probabilistic approaches, and translation resources ranged from bilingual dictionaries, parallel and comparable corpora to on-line MT systems and Wikipedia. Groups often used a combination of more than one resource. What is immediately noticeable in 2009 is that, although similarly to last year a number of different retrieval models were tested, there is a far more uniform approach to the translation problem.

Five of the ten groups that attempted cross-language tasks used the Google Translate functionality, while a sixth used the LEC Power Translator [14]. Another group also used an MT system combining it with concept-based techniques

but did not disclose the name of the MT system used [17]. The remaining three groups used a bilingual term list [18], a combination of resources including on-line and in house developed dictionaries [24], and Wikipedia translation links [19]. It should be noted that four out of the five groups in the bilingual to English and bilingual to French tasks and three out of five for the bilingual to German task used Google Translate, either on its own or in combination with another technique. One group reported that topic translation using a statistical MT system resulted in about 70% of the mean average precision (MAP) achieved when using Google Translate [25]. Another group [11] found that the results obtained by simply translating the query into all the target languages via Google gave results that were comparable to a far more complex strategy known as Cross-Language Explicit Semantic Analysis, CL-ESA, where the library catalog records and the queries are represented in a multilingual concept space that is spanned by aligned Wikipedia articles. As, overall, the CLEF2009 results were significantly better than those of CLEF 2008, can we take this as meaning that Google is going to solve the cross-language translation resource quandary?

Taking a closer look at three groups that did consistently well in the cross-language tasks we find the following. The group that had the top result for each of the three tasks was Chemnitz [16]. They also had consistently good monolingual results. Not surprisingly, they appear to have a very strong IR engine, which uses various retrieval models and combines the results. They used Snowball stemmers for English and French and an n-gram stemmer for German. They were one of the few groups that tried to address the multilinguality of the target collections. They used the Google service to translate the topic from the source language to the four most common languages in the target collections, queried the four indexes and combined the results in a multilingual result set. They found that their approach combining multiple indexed collections worked quite well for French and German but was disappointing for English.

Another group with good performance, Karlsruhe [17], also attempted to tackle the multilinguality of the collections. Their approach was again based on multiple indexes for different languages with rank aggregation to combine the different partial results. They ran language detectors on the collections to identify the different languages contained and translated the topics to the languages recognized. They used Snowball stemmers to stem terms in ten main languages, fields in other languages were not preprocessed. Disappointingly, a baseline consisting of a single index without language classification and a topic translated only to the index language achieved similar or even better results. For the translation step, they combined MT with a concept-based retrieval strategy based on Explicit Semantic Analysis and using the Wikipedia database in English, French and German as concept space.

A third group that had quite good cross-language results for all three collections was Trinity [12]. However, their monolingual results were not so strong. They used a language modelling retrieval paradigm together with a document re-ranking method which they tried experimentally in the cross-language context. Significantly, they also used Google Translate. Judging from the fact that

they did not do so well in the monolingual tasks, this seems to be the probable secret of their success for cross-language.

Of the three groups that submitted monolingual only runs, the INESC group achieved a consistently good performance, with the best MAP for the English collection and the third best for both French and German targets. They experimented an N-gram stemming technique together with query expansion and multinomial language modelling [23]. The Cuza group participated in the monolingual English task, using Lucene and addressing the multilingual aspect of the TEL collections by translating the title fields of the English topics into French and German, again using the Google API [22]. The third group, Opentext, focussed their attention on testing the stability and reusability of the test collections as reported above, rather than on the performance of their own retrieval system [10].

4 Persian@CLEF

This activity was again coordinated in collaboration with the Data Base Research Group (DBRG) of Tehran University. We were very disappointed that despite the fact that 14 groups registered for the CLEF 2009 Persian task, only four actually submitted results. And only one of these groups was from Iran. We suspect that one of the reasons for this was that the date for submission of results was not very convenient for the Iranian groups.

4.1 Tasks

The activity was organised as a typical ad hoc text retrieval task on newspaper collections. Two tasks were offered: monolingual retrieval; cross-language retrieval (English queries to Persian target) and 50 topics were prepared (see section 2.2). For each topic, participants had to find relevant documents in the collection and submit the results in a ranked list. Table 3 provides a breakdown of the number of participants and runs submitted by task and topic language.

4.2 Results

Table 6 shows the results for the two tasks, ordered by mean average precision. The table reports: the short name of the participating group; the mean average precision achieved by the experiment; the DOI of the experiment; and, where appropriate, the performance difference between the first and the last participant. Unfortunately, as can be seen in the table, something clearly went very wrong with the bilingual experiments and the results should probably be discounted.

Figure 11 compares the performances of the top participants of the Persian monolingual task.

Table 6. Best entries for the Persian tasks

Track	Rank	Participant	Experiment DOI	MAP
Monolingual	1st	jhu-apl	10.2415/AH-PERSIAN-MONO-FA-CLEF2009.JHU-APL.JHUFASK41R400TD	49.38%
	2nd	unine	10.2415/AH-PERSIAN-MONO-FA-CLEF2009.UNINE.UNINEPE4	49.37%
	3rd	opentext	10.2415/AH-PERSIAN-MONO-FA-CLEF2009.OPENTEXT.OTFA09TDE	39.53%
	4th	qazviniau	10.2415/AH-PERSIAN-MONO-FA-CLEF2009.QAZVINIAU.IAUPERFA3	37.62%
	5th	–	–	–%
	Difference			31.25%
Bilingual	1st	qazviniau	10.2415/AH-PERSIAN-BILI-X2FA-CLEF2009.QAZVINIAU.IAUPEREN3	2.72%
	2nd	–	–	–
	3rd	–	–	–
	4th	–	–	–
	5th	–	–	–
	Difference			–

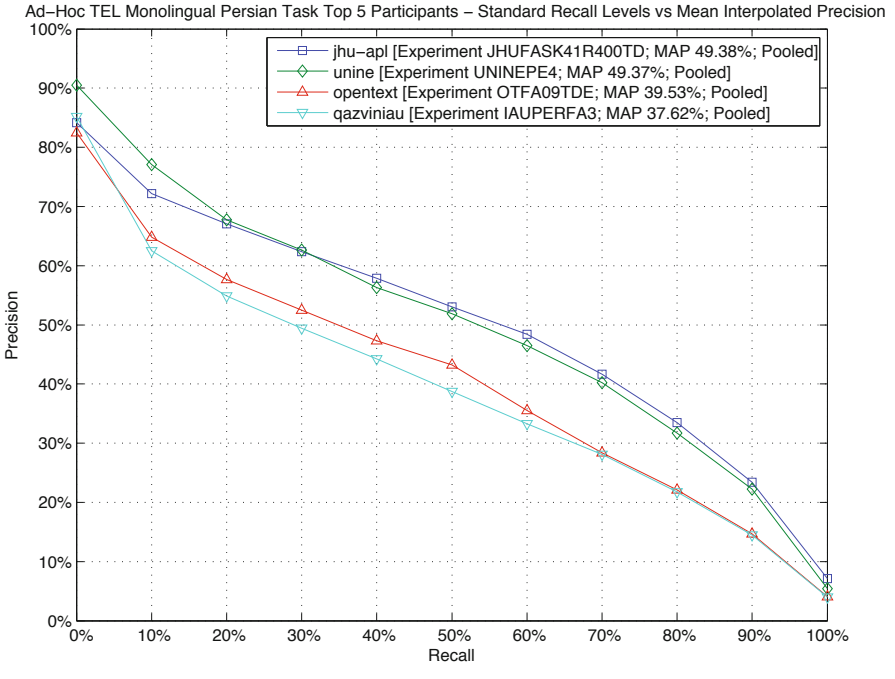


Fig. 11. Monolingual Persian

4.3 Approaches

As stated, only one group [20] attempted the bilingual task with the very poor results cited above. The technique they used was the same as that adopted for their bilingual to English experiments, exploiting Wikipedia translation links,

and the reason they give for the very poor performance here is that the coverage of Farsi in Wikipedia is still very scarce compared to that of many other languages.

In the monolingual Persian task, the top two groups had very similar performance figures. [26] found they had best results using a light suffix-stripping algorithm and by combining different indexing and searching strategies. In particular, they found that the use of blind query expansion could significantly improve retrieval effectiveness. Interestingly, their results this year do not confirm their findings for the same task last year when the use of stemming did not prove very effective [27]. The other group [15] tested variants of character n-gram tokenization; 4-grams, 5-grams, and skipgrams all provided about a 10% relative gain over plain words. The only Persian group focussed on testing a stemmer and light morphological analyser. Unlike [26] they found that blind relevance feedback hurt their precision [21].

An additional paper in these Proceedings, presents some post-campaign monolingual experiments [13]. These authors propose and test a variation of the vector space model which is based on phrases rather than single terms. They show a good precision for top-ranked documents when compared with other commonly used models.

5 Conclusions

In CLEF 2009 we deliberately repeated the TEL and Persian tasks offered in 2008 in order to build up our test collections. We are reasonably happy with the results for the TEL task: several groups worked on tackling the particular features of the TEL collections with varying success; evidence has been acquired on the effectiveness of a number of different IR strategies; there is a very strong indication of the validity of the Google Translate functionality.

On the other hand, the results for the Persian task were quite disappointing: very few groups participated; the results obtained are either in contradiction to those obtained previously and thus need further investigation [26] or tend to be a very straightforward repetition and confirmation of last year's results [15].

Acknowledgements

The TEL task was studied in order to provide useful input to The European Library (TEL); we express our gratitude in particular to Jill Cousins, Programme Director, and Sjoerd Siebinga, Technical Developer of TEL. Vivien Petras, Humboldt University, Germany, and Nicolas Moreau, Evaluation and Language Resources Distribution Agency, France, were responsible for the creation of the topics and the supervision of the relevance assessment work for the ONB and BNF data respectively. We thank them for their valuable assistance.

We should also like to acknowledge the enormous contribution to the coordination of the Persian task made by the Data Base Research group of the University of Tehran and in particular to Abolfazl AleAhmad and Hadi Amiri. They were

responsible for the preparation of the set of topics for the Hamshahri collection in Farsi and English and for the subsequent relevance assessments.

Least but not last, we would warmly thank Giorgio Maria Di Nunzio for all his immense contribution in carrying out the TEL and Persian tasks.

References

1. Agosti, M., Di Nunzio, G.M., Ferro, N.: The Importance of Scientific Data Curation for Evaluation Campaigns. In: Thanos, C., Borri, F. (eds.) DELOS Conference 2007 Working Notes, February 2007, pp. 185–193. ISTI-CNR, Gruppo ALI, Pisa, Italy (2007)
2. Alonso, O., Mizzaro, S.: Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In: Geva, S., Kamps, J., Peters, C., Sakai, T., Trotman, A., Voorhees, E. (eds.) Proc. SIGIR 2009 Workshop on The Future of IR Evaluation (2009), http://staff.science.uva.nl/~kamps/ireval/papers/paper_22.pdf
3. Braschler, M.: CLEF 2002 – Overview of Results. In: Braschler, M., Gonzalo, J., Kluck, M. (eds.) CLEF 2002. LNCS, vol. 2785, pp. 9–27. Springer, Heidelberg (2003)
4. Braschler, M., Peters, C.: CLEF 2003 Methodology and Metrics. In: Peters, C., Gonzalo, J., Braschler, M., Kluck, M. (eds.) CLEF 2003. LNCS, vol. 3237, pp. 7–20. Springer, Heidelberg (2004)
5. Cleverdon, C.W.: The Cranfield Tests on Index Languages Devices. In: Spärck Jones, K., Willett, P. (eds.) Readings in Information Retrieval, pp. 47–60. Morgan Kaufmann Publisher, Inc., San Francisco (1997)
6. Di Nunzio, G.M., Ferro, N.: Appendix A: Results of the TEL@CLEF Task. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/
7. Di Nunzio, G.M., Ferro, N.: Appendix B: Results of the Persian@CLEF Task. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/
8. Sanderson, M., Joho, H.: Forming Test Collections with No System Pooling. In: Sanderson, M., Järvelin, K., Allan, J., Bruza, P. (eds.) Proc. 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004), pp. 33–40. ACM Press, New York (2004)
9. Tomlinson, S.: Sampling Precision to Depth 10000 at CLEF 2008. In: Peters, C., et al. (eds.) Evaluating Systems for Multilingual and Multimodal Information Access. LNCS, vol. 5706, pp. 163–169. Springer, Heidelberg (2009)
10. Tomlinson, S.: Sampling Precision to Depth 10000 at CLEF 2009. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 78–85. Springer, Heidelberg (2010)
11. Anderka, M., Lipka, N., Stein, B.: Evaluating Cross-Language Explicit Semantic Analysis and Cross Querying. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 50–57. Springer, Heidelberg (2010)
12. Zhou, D., Wade, V.: Smoothing Methods and Cross-Language Document Re-Ranking. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 62–69. Springer, Heidelberg (2010)
13. Habibian, A., AleAhmad, A., Shakery, A.: Ad Hoc Information Retrieval for Perisan. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 110–119. Springer, Heidelberg (2010)

14. Larson, R.R.: Multilingual Query Expansion for CLEF Adhoc-TEL. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 86–89. Springer, Heidelberg (2010)
15. McNamee, P.: JHU Experiments in Monolingual Farsi Document Retrieval at CLEF 2009. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop 2009, http://www.clef-campaign.org/
16. Kuersten, J.: Chemnitz at CLEF 2009 Ad-Hoc TEL Task: Combining Different Retrieval Models and Addressing the Multilinguality. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop 2009, http://www.clef-campaign.org/
17. Sorg, P., Braun, M., Nicolay, D., Cimiano, P.: Cross-lingual Information Retrieval based on Multiple Indexes. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/
18. Katsiouli, P., Kalamboukis, T.: An Evaluation of Greek-English Cross Language Retrieval within the CLEF Ad-Hoc Bilingual Task. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/
19. Jadidinejad, A.H., Mahmoudi, F.: Cross-Language Information Retrieval Using Meta-Language Index Construction and Structural Queries. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 70–77. Springer, Heidelberg (2010)
20. Jadidinejad, A.H., Mahmoudi, F.: Query Wikification: Mining Structured Queries from Unstructured Information Needs using Wikipedia-based Semantic Analysis. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/
21. Jadidinejad, A.H., Mahmoudi, F.: PerStem: A Simple and Efficient Stemming Algorithm for Persian Language. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 98–101. Springer, Heidelberg (2010)
22. Iftne, A., Mihaila, A.-E., Epure, I.-P.: UAIC: Participation in TEL@CLEF task. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/
23. Machado, J., Martins, B., Borbinha, J.: Experiments with N-Gram Prefixes on a Multinomial Language Model versus Lucene's off-the-shelf ranking scheme and Rocchio Query Expansion (TEL@CLEF Monolingual Task). In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 90–97. Springer, Heidelberg (2010)
24. Bosca, A., Dini, L.: CACAO Project at the TEL@CLEF 2009 Task. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/
25. Leveling, J., Zhou, D., Jones, G.F., Wade, V.: Document Expansion, Query Translation and Language Modeling for Ad-hoc IR. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 58–61. Springer, Heidelberg (2010)
26. Dolamic, L., Savoy, J.: Ad Hoc Retrieval with the Persian Language. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 102–109. Springer, Heidelberg (2010)
27. Dolamic, L., Fautsch, C., Savoy, J.: UniNE at CLEF 2008: TEL and Persian IT. In: Peters, C., et al. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 178–185. Springer, Heidelberg (2009)

CLEF 2009 Ad Hoc Track Overview: Robust-WSD Task

Eneko Agirre[1], Giorgio Maria Di Nunzio[2], Thomas Mandl[3], and Arantxa Otegi[1]

[1] Computer Science Department, University of the Basque Country, Spain
{e.agirre,arantza.otegi}@ehu.es
[2] Department of Information Engineering, University of Padua, Italy
dinunzio@dei.unipd.it
[3] Information Science, University of Hildesheim, Germany
mandl@uni-hildesheim.de

Abstract. The Robust-WSD at CLEF 2009 aims at exploring the contribution of Word Sense Disambiguation to monolingual and multilingual Information Retrieval. The organizers of the task provide documents and topics which have been automatically tagged with Word Senses from WordNet using several state-of-the-art Word Sense Disambiguation systems. The Robust-WSD exercise follows the same design as in 2008. It uses two languages often used in previous CLEF campaigns (English, Spanish). Documents were in English, and topics in both English and Spanish. The document collections are based on the widely used LA94 and GH95 news collections. All instructions and datasets required to replicate the experiment are available from the organizers website (http://ixa2.si.ehu.es/clirwsd/). The results show that some top-scoring systems improve their IR and CLIR results with the use of WSD tags, but the best scoring runs do not use WSD.

1 Introduction

The Robust-WSD task at CLEF 2009 aims at exploring the contribution of Word Sense Disambiguation to monolingual and multilingual Information Retrieval. The organizers of the task provide documents and topics which have been automatically tagged with Word Senses from WordNet using several state-of-the-art Word Sense Disambiguation systems. The task follows the same design as at CLEF 2008.

The robust task ran for the fourth time at CLEF 2009. It is an Ad-Hoc retrieval task based on data of previous CLEF campaigns. The robust task emphasizes the difficult topics by a non-linear integration of the results of individual topics into one result for a system, using the geometric mean of the average precision for all topics (GMAP) as an additional evaluation measure [17,18]. Given the difficulty of the task, training data including topics and relevance assessments was provided for the participants to tune their systems to the collection.

For the second year, the robust task also incorporated word sense disambiguation information provided by the organizers to the participants. The task follows

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 36–49, 2010.

the 2007 joint SemEval-CLEF task [2] and the 2008 Robust-WSD exercise [3], and has the aim of exploring the contribution of word sense disambiguation to monolingual and cross-language information retrieval. The goal of the task is to test whether WSD can be used beneficially for retrieval systems, and thus participants were required to submit at least one baseline run without WSD and one run using the WSD annotations. Participants could also submit four further baseline runs without WSD and four runs using WSD.

The experiment involved both monolingual (topics and documents in English) and bilingual experiments (topics in Spanish and documents in English). In addition to the original documents and topics, the organizers of the task provided both documents and topics which had been automatically tagged with word senses from WordNet version 1.6 using two state-of-the-art word sense disambiguation systems, UBC [1] and NUS [8]. These systems provided weighted word sense tags for each of the nouns, verbs, adjectives and adverbs that they could disambiguate. These systems participated in the Semeval 2007 task on Word Sense Disambiguation [16], with similar results. NUS ranked 4th in the all-words task, with an accuracy of 57.4, and UBC ranked 5th, with an accuracy of 54.4. In the all-words task, the output of both systems was compared with the gold stadard sense tags on a sample of three documents.

In addition, the participants could use publicly available data from the English and Spanish wordnets in order to test different expansion strategies. Note that given the tight alignment of the Spanish and English wordnets, the wordnets could also be used to translate directly from one sense to another, and perform expansion to terms in another language.

The datasets used in this task can be used in the future to run further experiments. Check `http://ixa2.si.ehu.es/clirwsd` for information of how to access the datasets. Topics and relevance judgements are freely available. The document collection can be obtained from ELDA purchasing the CLEF Test Suite for the CLEF 2000-2003 Campaigns – Evaluation Package. As an alternative, the website offers the unordered set of words in each document, that is, the full set of documents where the positional information has been eliminated to avoid replications of the originals. Lucene indexes for the later are also available from the website.

In this paper, we first present the task setup, the evaluation methodology and the participation in the different tasks (Section 2). We then describe the main features of each task and show the results (Sections 3 - 5). The final section provides a brief summing up. For information on the various approaches and resources used by the groups participating in this task and the issues they focused on, we refer the reader to the rest of the papers in the Robust-WSD part of the Ad Hoc section of these Proceedings.

2 Task Setup

The Ad Hoc task in CLEF adopts a corpus-based, automatic scoring method for the assessment of system performance, based on ideas first introduced in

the Cranfield experiments in the late 1960s [9]. The **tasks** offered are studied in order to effectively measure textual document retrieval under specific conditions. The **test collections** are made up of **documents**, **topics** and **relevance assessments**. The topics consist of a set of statements simulating information needs from which the systems derive the queries to search the document collections. Evaluation of system performance is then done by judging the documents retrieved in response to a topic with respect to their relevance, and computing the recall and precision measures.

2.1 Test Collections

The Documents. The robust task used existing CLEF news collections but with word sense disambiguation (WSD) information added. The word sense disambiguation data was automatically added by systems from two leading research laboratories, UBC [1] and NUS [8]. Both systems returned word senses from the English WordNet, version 1.6.

The document collections were offered both with and without WSD, and included the following[1]:

- LA Times 94 (with word sense disambiguated data); ca 113,000 documents, 425 MB without WSD, 1,448 MB (UBC) or 2,151 MB (NUS) with WSD;
- Glasgow Herald 95 (with word sense disambiguated data); ca 56,500 documents, 154 MB without WSD, 626 MB (UBC) or 904 MB (NUS) with WSD.

The Topics. Topics are structured statements representing information needs. Each topic typically consists of three parts: a brief title statement; a one-sentence description; a more complex narrative the relevance assessment criteria. Topics are prepared in xml format and identified by means of a Digital Object Identifier (DOI)[2] of the experiment [14] which allows us to reference and cite them.

The WSD robust task used existing CLEF topics in English and Spanish as follows:

- CLEF 2001; Topics 10.2452/41-AH – 10.2452/90-AH; LA Times 94
- CLEF 2002; Topics 10.2452/91-AH – 10.2452/140-AH; LA Times 94
- CLEF 2003; Topics 10.2452/141-AH – 10.2452/200-AH; LA Times 94, Glasgow Herald 95
- CLEF 2004; Topics 10.2452/201-AH – 10.2452/250-AH; Glasgow Herald 95
- CLEF 2005; Topics 10.2452/251-AH – 10.2452/300-AH; LA Times 94, Glasgow Herald 95
- CLEF 2006; Topics 10.2452/301-AH – 10.2452/350-AH; LA Times 94, Glasgow Herald 95

[1] A sample document and dtd are available at http://ixa2.si.ehu.es/clirwsd/
[2] http://www.doi.org/

```
<top>
    <num>10.2452/141-WSD-AH</num>

    <EN-title>
        <TERM ID="10.2452/141-WSD-AH-1" LEMA="letter" POS="NNP">
            <WF>Letter</WF>
            <SYNSET SCORE="0" CODE="05115901-n"/>
            <SYNSET SCORE="0" CODE="05362432-n"/>
            <SYNSET SCORE="0" CODE="05029514-n"/>
            <SYNSET SCORE="1" CODE="04968965-n"/>
        </TERM>

        <TERM ID="10.2452/141-WSD-AH-2" LEMA="bomb" POS="NNP">
            <WF>Bomb</WF>
            <SYNSET SCORE="0.888888888888889" CODE="02310834-n"/>
            <SYNSET SCORE="0" CODE="05484679-n"/>
            <SYNSET SCORE="0.111111111111111" CODE="02311368-n"/>
        </TERM>

        <TERM ID="10.2452/141-WSD-AH-3" LEMA="for" POS="IN">
            <WF>for</WF>
        </TERM>

        ...

    </EN-title>

    <EN-desc>
        <TERM ID="10.2452/141-WSD-AH-5" LEMA="find" POS="VBP">
            <WF>Find</WF>
            <SYNSET SCORE="0" CODE="00658116-v"/>

            ...

        </TERM>

        ...

    </EN-desc>

    <EN-narr>
        ...
    </EN-narr>
</top>
```

Fig. 1. Example of Robust WSD topic: topic `10.2452/141-WSD-AH`

Topics from years 2001, 2002 and 2004 were used as training topics (relevance assessments were offered to participants), and topics from years 2003, 2005 and 2006 were used for the test.

All topics were offered both with and without WSD. Topics in English were disambiguated by both UBC [1] and NUS [8] systems, yielding word senses from WordNet version 1.6. A large-scale disambiguation system for Spanish was not available, so we used the first-sense heuristic, yielding senses from the Spanish wordnet, which is tightly aligned to the English WordNet version 1.6 (i.e., they share synset numbers or sense codes). An excerpt from a topic is shown in Figure 1, where each term in the topic is followed by its senses with their respective scores as assigned buy the automatic WSD system[3].

[3] Full sample and dtd are available at `http://ixa2.si.ehu.es/clirwsd/`

Relevance Assessment. The number of documents in large test collections such as CLEF makes it impractical to judge every document for relevance. Instead approximate recall values are calculated using pooling techniques. The robust WSD task used existing relevance assessments from previous years. The relevance assessments regarding the training topics were provided to participants before competition time.

The total number of assessments was 66,441 documents of which 4,327 were relevant. The distribution of the pool according to each year was the following:

- CLEF 2003: 23,674 documents, 1,006 relevant;
- CLEF 2005: 19,790 document, 2,063 relevant;
- CLEF 2006: 21,247 document, 1,258 relevant;

Seven topics had no relevant documents at all: 10.2452/149-AH, 10.2452/161-AH, 10.2452/166-AH, 10.2452/186-AH, 10.2452/191-AH, 10.2452/195-AH, 10.2452/321-AH. Each topic had an average of about 28 relevant documents and a standard deviation of 34, a minimum of 1 relevant document and a maximum of 229 relevant documents per topic.

2.2 Result Calculation

Evaluation campaigns such as TREC and CLEF are based on the belief that the effectiveness of *Information Retrieval Systems (IRSs)* can be objectively evaluated by an analysis of a representative set of sample search results. For this, effectiveness measures are calculated based on the results submitted by the participants and the relevance assessments. Popular measures usually adopted for exercises of this type are Recall and Precision. Details on how they are calculated for CLEF are given in [7].

The robust task emphasizes the difficult topics by a non-linear integration of the results of individual topics into one result for a system, using the geometric mean of the average precision for all topics (GMAP) as an additional evaluation measure [17,18]. This makes especially sense in multilingual retrieval where results can differ from results based on MAP [15].

The individual results for all official Ad Hoc experiments in CLEF 2009 are given in the one of the Appendices of the CLEF 2009 Working Notes [11].

2.3 Participants and Experiments

As shown in Table 1, 10 groups submitted 89 runs for the Robust tasks:

- 8 groups submitted monolingual non-WSD runs (25 runs out of 89);
- 5 groups also submitted bilingual non-WSD runs (13 runs out of 89).

All groups submitted WSD runs (51 out of 89 runs):

- 10 groups submitted monolingual WSD runs (33 out of 89 runs)
- 5 groups submitted bilingual WSD runs (18 out of 89 runs)

Table 1. CLEF 2009 Ad Hoc Robust participants. See text in Section 2.3. for comments on participants with *.

participant	task	No. experiments
alicante*	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	3
darmstadt	AH-ROBUST-MONO-EN-TEST-CLEF2009	5
darmstadt	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	5
geneva*	AH-ROBUST-MONO-EN-TEST-CLEF2009	5
geneva*	AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009	1
geneva*	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	2
ixa	AH-ROBUST-BILI-X2EN-TEST-CLEF2009	1
ixa	AH-ROBUST-MONO-EN-TEST-CLEF2009	1
ixa	AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009	4
ixa	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	3
jaen*	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	2
know-center	AH-ROBUST-BILI-X2EN-TEST-CLEF2009	3
know-center	AH-ROBUST-MONO-EN-TEST-CLEF2009	3
know-center	AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009	3
know-center	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	3
reina*	AH-ROBUST-BILI-X2EN-TEST-CLEF2009	5
reina*	AH-ROBUST-MONO-EN-TEST-CLEF2009	5
reina*	AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009	5
reina*	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	5
ufrgs	AH-ROBUST-BILI-X2EN-TEST-CLEF2009	1
ufrgs	AH-ROBUST-MONO-EN-TEST-CLEF2009	1
ufrgs	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	1
uniba	AH-ROBUST-BILI-X2EN-TEST-CLEF2009	3
uniba	AH-ROBUST-MONO-EN-TEST-CLEF2009	3
uniba	AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009	5
uniba	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	5
valencia	AH-ROBUST-MONO-EN-TEST-CLEF2009	2
valencia	AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009	4

Table 2 provides a breakdown of the number of participants and submitted runs by task. Note that jaen submitted a monolingual non-WSD run as if it was a WSD run, and that alicante missed to send their non-WSD run on time. Although REINA submitted some runs under WSD, they did not use WSD information [20], only lemma and PoS. Geneva did not submit a paper describing their systems. The figures used in this paper are the official figures at the time of the task.

3 Results

Table 3 shows the best results for the monolingual runs, and Table 4 shows the best results for the bilingual runs. In the following pages, Figure 2 shows the performances of the best systems in terms of average precision of the top participants of the Robust Monolingual and Monolingual WSD, and Figure 3

Table 2. Number of runs per track

Track	# Part.	# Runs
Robust Mono English Test	8	25
Robust Mono English Test WSD	10	33
Robust Biling. English Test	5	13
Robust Biling. English Test WSD	5	18

Table 3. Best entries for the robust monolingual task, including both WSD and non-WSD runs. The **Q** columns shows the information used to build the query.

	Rank	Participant	Q	Experiment DOI	MAP	GMAP
Non-WSD	**1st**	darmstadt	TD	10.2415/AH-ROBUST-MONO-EN-TEST-CLEF2009.DARMSTADT.DA_4	45.09%	20.42%
	2nd	reina	TDN	10.2415/AH-ROBUST-MONO-EN-TEST-CLEF2009.REINA.ROB2	44.52%	21.18%
	3rd	uniba	TDN	10.2415/AH-ROBUST-MONO-EN-TEST-CLEF2009.UNIBA.UNIBAKRF	42.50%	17.93%
	4th	geneva	TDN	10.2415/AH-ROBUST-MONO-EN-TEST-CLEF2009.GENEVA.ISIENNATTDN	41.71%	17.88%
	5th	know-center	TD	10.2415/AH-ROBUST-MONO-EN-TEST-CLEF2009.KNOW-CENTER.ASSO	41.70%	18.64%
WSD	**1st**	darmstadt	TD	10.2415/AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009.DARMSTADT.DA_WSD_4	45.00%	20.49%
	2nd	uniba	TDN	10.2415/AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009.UNIBA.UNIBAKEYSYNRF	43.46%	19.60%
	3rd	know-center	TD	10.2415/AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009.KNOW-CENTER.ASSOWSD	42.22%	19.47%
	4th	geneva	TDN	10.2415/AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009.GENEVA.ISINUSLWTDN	38.11%	16.26%
	5th	ixa	TD	10.2415/AH-ROBUST-WSD-MONO-EN-TEST-CLEF2009.IXA.ENENBESTSENSE500DOCS	38.05%	16.57%

Table 4. Best entries for the robust ES-EN bilingual task, including both WSD and non-WSD runs. The **Q** columns shows the information used to build the query.

	Rank	Participant	Q	Experiment DOI	MAP	GMAP
Non-WSD	**1st**	reina	TDN	10.2415/AH-ROBUST-BILI-X2EN-TEST-CLEF2009.REINA.BILI2	38.42%	15.11%
	2nd	uniba	TDN	10.2415/AH-ROBUST-BILI-X2EN-TEST-CLEF2009.UNIBA.UNIBACROSSKEYRF	38.09%	13.11%
	3rd	know-center	TD	10.2415/AH-ROBUST-BILI-X2EN-TEST-CLEF2009.KNOW-CENTER.BILIASSO	28.98%	06.79%
	4th	ufrgs	TD	10.2415/AH-ROBUST-BILI-X2EN-TEST-CLEF2009.UFRGS.BILINGUAL	27.65%	07.37%
	5th	ixa	TD	10.2415/AH-ROBUST-BILI-X2EN-TEST-CLEF2009.IXA.ESENNOWSD	18.05%	01.90%
WSD	**1st**	uniba	TDN	10.2415/AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009.UNIBA.UNIBACROSSKEYSYNRF	37.53%	13.82%
	2nd	geneva	TD	10.2415/AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009.GENEVA.ISINUSWSDTD	36.63%	16.02%
	3rd	know-center	TD	10.2415/AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009.KNOW-CENTER.BILIASSOWSD	29.64%	07.05%
	4rd	ixa	TD	10.2415/AH-ROBUST-WSD-BILI-X2EN-TEST-CLEF2009.IXA.ESEN1STTOPSBESTSENSE500DOCS	18.38%	01.98%

shows the performances of the best participants of the Robust Bilingual and Bilingual WSD. Some teams used the Title and Description fields to construct the query (TD), while others also used the narrative (TDN).

The comparison of the bilingual runs with respect to the monolingual results yield the following:

- ES → EN: 85.2% of best monolingual English IR system (MAP);
- ES → EN WSD: 83.3% of best monolingual English IR system (MAP);

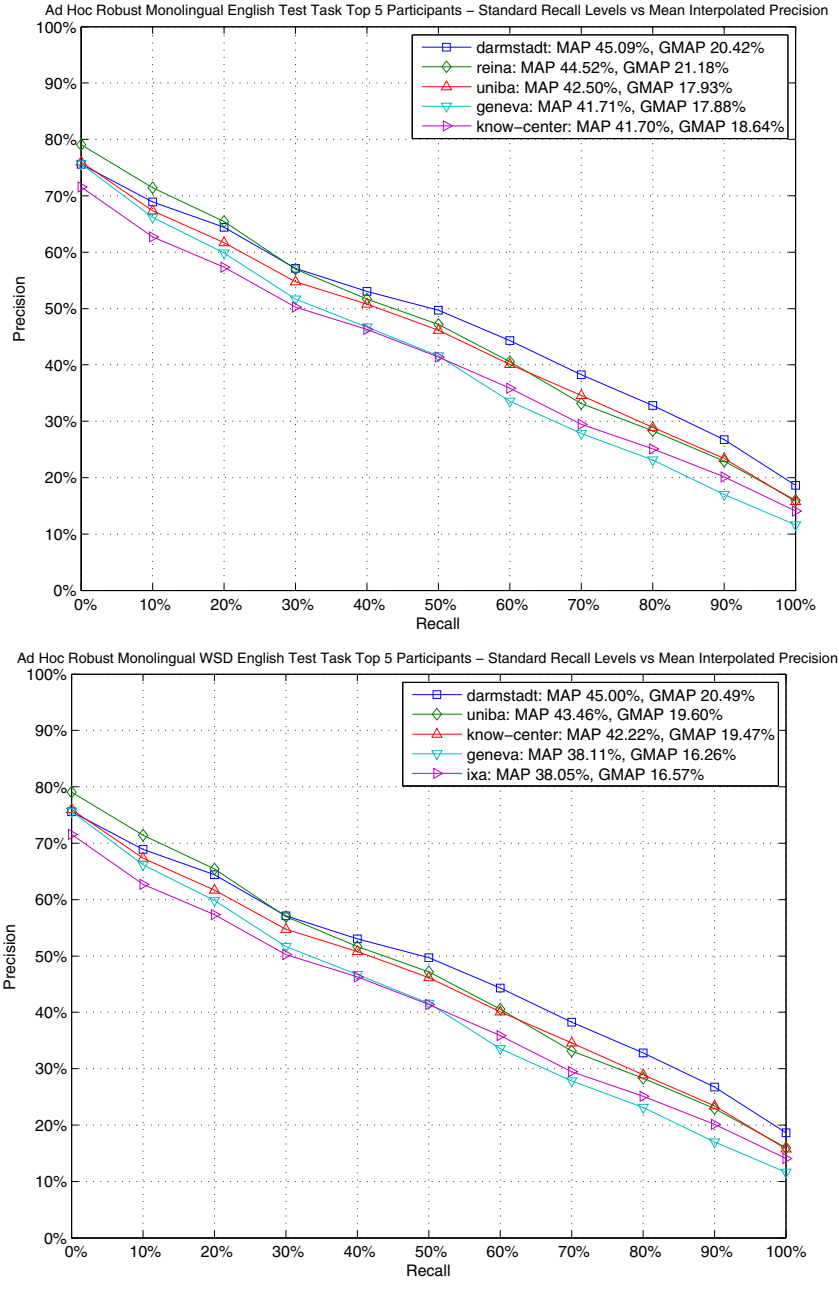


Fig. 2. Mean average precision of the top 5 participants of the Robust Monolingual English Task (top graph) and Robust WSD Monolingual English Task (bottom).

Table 5. Statistical tests comparison between non-WSD and WSD runs. Differences or equalities are statistically significant with alpha = 5%

	Monolingual	Bilingual
Task	Non-WSD > WSD	Non-WSD > WSD
Set of best runs	Non-WSD > WSD	Non-WSD = WSD
Single best run	Non-WSD = WSD	Non-WSD = WSD

3.1 Statistical Testing

When the goal is to validate how well results can be expected to hold beyond a particular set of queries, statistical testing can help to determine what differences between runs appear to be real as opposed to differences that are due to sampling issues. We aim to identify whether the results of the runs of a task are significantly different from the results of other tasks. In particular, we want to test whether there is any difference between applying WSD techniques or not. Significantly different in this context means that the difference between the performance scores for the runs in question appears greater than what might be expected by pure chance. As with all statistical testing, conclusions will be qualified by an error probability, which was chosen to be 0.05 in the following. We have designed our analysis to follow closely the methodology used by similar analyses carried out for Text REtrieval Conference (TREC) [18].

We used the MATLAB Statistics Toolbox, which provides the necessary functionality plus some additional functions and utilities. Following the approach presented by [10], the first step is to verify whether the distributions of performances are normal, the second step is to analyze whether the variances of the distributions are equal, and finally to test whether the means of the distributions are the same. Three different pairs of distributions were analyzed to verify the differences between WSD and non-WSD experiments:

- Task:
  - Robust Monolingual vs Robust WSD Monolingual;
  - Robust Bilingual vs Robust WSD Bilingual.
- Set of best experiments:
  - Best performers of Robust Monolingual vs Best Robust WSD Monolingual (experiments of Table 3);
  - Best performers of Robust Bilingual vs Best performers Robust WSD Bilingual (experiments of Table 4).
- Single best experiment:
  - Best Robust Monolingual vs Best Robust WSD Monolingual;
  - Best Robust Bilingual vs Best Robust WSD Bilingual.

Results are summarized in Table 3.1, showing that overall, systems not using WSD perform better than those using WSD. When we take the best systems, results are not statistically different. These comparisons are done without taking into account who is producing which runs. Another alternative is to analize each participant system separately, as we will do in the next section.

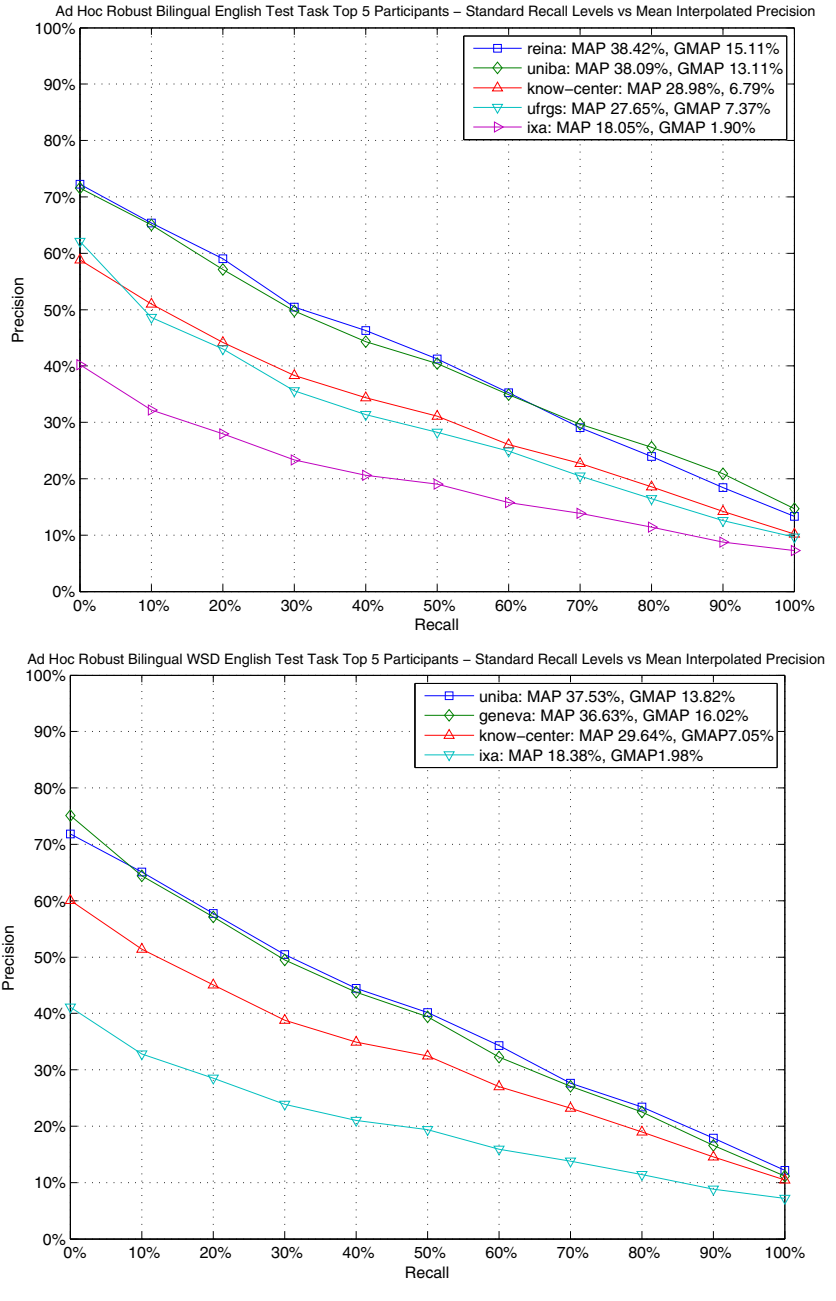


Fig. 3. Mean average precision of the top 5 participants of the Robust Bilingual English Task (top graph) and Robust WSD Bilingual English Task (bottom)

Table 6. Statistical tests comparison between non-WSD and WSD runs of best participants. Differences or equalities are statistically significant with alpha = 5%.

System	Monolingual	Bilingual
darmstadt	Non-WSD = WSD	n/a
uniba	Non-WSD < WSD	Non-WSD = WSD
geneva	Non-WSD > WSD	n/a
know-center	Non-WSD = WSD	Non-WSD = WSD

3.2 Analysis

In this section we focus on the comparison between WSD and non-WSD runs of each participant. Overall, the best MAP and GMAP results in the monolingual system were for two distinct runs which did not use WSD information, but several participants were able to obtain their best MAP and GMAP scores using WSD information. In the bilingual experiments, the best results in MAP were for non-WSD runs, but two participants were able to profit from the WSD annotations. As it is difficult to summarize the behavior of all participants, we will only mention the performance of the best teams, as given in Tables 3 and 4. In addition, Table 3.2 summarizes whether the best WSD and non-WSD runs for participants that submitted both runs are statistically significant. The interested reader is directed to the papers of each participant in this volume for additional details.

In the **monolingual experiments**, cf. Table 3, the best results overall in MAP were for **darmstadt**. Their WSD runs scored very similar to the non-WSD runs, with a slight decrease of MAP (0.09 percentage points, with no statistical difference) and a slight increase of GMAP (0.07 percentage points) [19]. The method to include WSD information was to create additional indexes for word senses, and then combine them with other indexes using weights as optimized from training. The retrieval system used the BM25 model, with an additional monolingual translation-based model.

The second best MAP score and best GMAP was attained by **reina** [20] without WSD. Unfortunately they did not submit any run using WSD. Systems such as this introduce noise in the comparisons in the previous section.

The third best MAP and second GMAP where obtained by **uniba** [5] using WSD. This team showed a 0.94 statistically different increase in MAP and 1.67 increase in GMAP with respect to their best non-WSD run. They constructed an additional index with synset numbers, and then combined the synsets using the N-levels model.

geneva [12] also attained good results, but their WSD system had a statistically significant drop in both MAP and GMAP. Unfortunately, the authors did not explain how they integrated WSD information in their system.

Finally, **know-center** [13] attained 0.52 improvements in MAP using WSD (not statistically significant difference) and 0.83 increase in GMAP with the use of WSD. They added synsets and synonyms to the index, and used an axiomatic retrieval approach.

In the **bilingual experiments**, cf. Table 4, the best results overall in MAP were for **reina** with a system which did not use WSD annotations [20], and again, they did not submit runs using WSD.

The best GMAP was for **geneva** using WSD [12], but unfortunately, they did not submit any non-WSD run.

uniba [5] got the second best MAP, with better MAP for the non-WSD run and better GMAP for the WSD run. The differences were small in both cases (0.56 in MAP, 0.71 in GMAP).

Those three teams had the highest results, well over 35% MAP, and the rest got more modest performances. **know-center** [13] reported better results using WSD information (0.66 MAP, 0.26 GMAP). **Ufrgs** [6] only submitted the WSD result. Finally **ixa** [4] got low results, with small improvements using WSD information (0.33 MAP, 0.08 GMAP).

All in all, the exercise showed that some teams did improve results using WSD (close to 1 MAP point and more than 1 GMAP point in monolingual, and below 1 MAP/GMAP point in bilingual), but the best results for both monolingual and bilingual tasks were for systems which did not use WSD.

4 Conclusions

This new edition of the robust WSD exercise has measured to what extent IR systems could profit from automatic word sense disambiguation information. The conclusions on the monolingual subtask are similar to the conclusions of 2008. The evidence for using WSD in monolingual IR is mixed. Some top scoring groups report improvements in MAP and GMAP, with significant improvements in the case of **uniba**, which attained the third best results. Still, the best overall scores are for two systems not using WSD. Regarding the cross-lingual task, the situation is very similar, but the reported improvements using WSD are smaller.

The lower performance of some groups when using WSD seems to indicate that using WSD for IR is not straightforward, and can lead to worse results if not done with care. From another perspective, the results of groups which do attest significant improvements are very relevant, as they show that a careful system design can render WSD information effective. We thus think that, overall, the results of the 2008 and 2009 campaigns are promising, and that there is still room for improvement.

The instructions and all datasets to replicate the results (including Lucene indexes) are available from `http://ixa.si.ehu.es/clirwsd`. Topics and relevance judgements are freely available. The document collection can be obtained from ELDA purchasing the CLEF Test Suite for the CLEF 2000-2003 Campaigns – Evaluation Package. As an alternative, the website offers the unordered set of words in each document, that is, the full set of documents where the positional information has been eliminated to avoid replications of the originals. Lucene indexes for the later are also available from the website. Given the availability of these resources, interested parties can now evaluate their own systems, and we thus felt that there is no need to organize another edition of the competition.

Acknowledgements

The robust task was partially funded by the Ministry of Education (project KNOW2 TIN2009-14715) and the European Commission (project KYOTO ICT-2007-211423). We want to thank Oier Lopez de Lacalle, who run the UBC WSD system, and Yee Seng Chan, Hwee Tou Ng and Zhi Zhong, who run the NUS WSD system. Their generous contribution was invaluable to organize this exercise.

References

1. Agirre, E., Lopez de Lacalle, O.: UBC-ALM: Combining k-NN with SVD for WSD. In: In Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), Czech Republic, Prague, pp. 341–345 (2007)
2. Agirre, E., Magnini, B., Lopez de Lacalle, O., Otegi, A., Rigau, G., Vossen, P.: SemEval-2007 Task01: Evaluating WSD on Cross-Language Information Retrieval. In: Carol, P., et al. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 908–917. Springer, Heidelberg (2008)
3. Agirre, E., Di Nunzio, G.M., Ferro, N., Peters, C., Mandl, T.: CLEF 2008: Ad Hoc Track Overview. In: Carol, P., et al. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 118–125. Springer, Heidelberg (2009)
4. Agirre, E., Otegi, A., Zaragoza, H.: Using Semantic Relatedness and Word Sense Disambiguation for (CL)IR. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 166–173. Springer, Heidelberg (2010)
5. Basile, P., Caputo, A., Semeraro, G.: UNIBA-SENSE at CLEF 2009: Robust WSD task. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 150–157. Springer, Heidelberg (2010)
6. Borges, T.B., Moreira, V.P.: UFRGS@CLEF2009: Retrieval by Numbers. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 135–141. Springer, Heidelberg (2010)
7. Braschler, M., Peters, C.: CLEF 2003 Methodology and Metrics. In: Peters, C., Gonzalo, J., Braschler, M., Kluck, M. (eds.) CLEF 2003. LNCS, vol. 3237, pp. 7–20. Springer, Heidelberg (2004)
8. Chan, Y.S., Ng, H.T., Zhong, Z.: NUS-PT: Exploiting Parallel Texts for Word Sense Disambiguation in the English All-Words Tasks. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), Prague, Czech Republic, pp. 253–256 (2007)
9. Cleverdon, C.: The Cranfield Tests on Index Language Devices. In: Sparck Jones, K., Willett, P. (eds.) Readings in Information Retrieval, pp. 47–59. Morgan Kaufmann Publisher, Inc., San Francisco (1997)
10. Crivellari, F., Di Nunzio, G.M., Ferro, N.: A statistical and graphical methodology for comparing bilingual to monolingual cross-language information retrieval. In: Agosti, M. (ed.) Information Access through Search Engines and Digital Libraries. Information Retrieval Series, vol. 22, pp. 171–188. Springer, Heidelberg (2008)
11. Di Nunzio, G.M., Ferro, N.: Appendix C: Results of the Robust Task. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/

12. Guyot, J., Falquet, G., Radhouani, S.: UniGe at CLEF 2009 Robust WSD Task. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/
13. Kern, R., Juffinger, A., Granitzer, M.: Application of Axiomatic Approaches to Crosslanguage Retrieval. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 142–149. Springer, Heidelberg (2010)
14. Paskin, N. (ed.): The DOI Handbook – Edition 4.4.1. International DOI Foundation, IDF (2006), http://dx.doi.org/10.1000/186
15. Mandl, T., Womser-Hacker, C., Di Nunzio, G.M., Ferro, N.: How Robust are Multilingual Information Retrieval Systems? In: Proceedings ACM SAC Symposium on Applied Computing (SAC), Fortaleza, Brazil, March 16-20, pp. 1132–1136 (2008)
16. Pradhan, S., Loper, E., Dligach, D., Palmer, M.: SemEval Task-17: English Lexical Sample, SRL and All Words. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, pp. 87–92. Association for Computational Linguistics (2007)
17. Robertson, S.: On GMAP: and Other Transformations. In: Yu, P.S., Tsotras, V., Fox, E.A., Liu, C.B. (eds.) Proc. 15th International Conference on Information and Knowledge Management (CIKM 2006), pp. 78–83. ACM Press, New York (2006)
18. Voorhees, E.M.: The TREC Robust Retrieval Track. SIGIR Forum 39, 11–20 (2005)
19. Wolf, E., Bernhard, D., Gurevych, I.: Combining Probabilistic and Translation-Based Models for Information Retrieval based on Word Sense Annotations Information Retrieval. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 120–127. Springer, Heidelberg (2010)
20. Zazo, A., Figuerola, C.G., Alonso Berrocal, J.L., Gomez, R.: REINA at CLEF 2009 Robust-WSD Task: Partial Use of WSD Information for Retrieval. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009), http://www.clef-campaign.org/

Evaluating Cross-Language Explicit Semantic Analysis and Cross Querying

Maik Anderka, Nedim Lipka, and Benno Stein

Bauhaus-Universität Weimar, 99421 Weimar, Germany
maik.anderka@uni-weimar.de

Abstract. This paper describes our participation in the TEL@CLEF task of the CLEF 2009 ad-hoc track. The task is to retrieve items from various multilingual collections of library catalog records, which are relevant to a user's query. Two different strategies are employed: (*i*) the Cross-Language Explicit Semantic Analysis, CL-ESA, where the library catalog records and the queries are represented in a multilingual concept space that is spanned by aligned Wikipedia articles, and, (*ii*) a Cross Querying approach, where a query is translated into all target languages using Google Translate and where the obtained rankings are combined. The evaluation shows that both strategies outperform the monolingual baseline and achieve comparable results.

Furthermore, inspired by the Generalized Vector Space Model we present a formal definition and an alternative interpretation of the CL-ESA model. This interpretation is interesting for real-world retrieval applications since it reveals how the computational effort for CL-ESA can be shifted from the query phase to a preprocessing phase.

1 Introduction

Cross-language information retrieval, CLIR, is the task of retrieving documents from a target collection written in a language different from the language of a user's query. CLIR systems give multilingual users the possibility to express queries in any language, e.g., their native language, and to obtain result documents in all languages they are familiar with. Since CLIR is not restricted to collections in the query language more sources can be included in the retrieval process, and the chance to fulfill a particular information need of a multilingual user is higher. Another use case for CLIR techniques is cross-language plagiarism detection, where the query corresponds to a suspicious document and the target collection is a reference corpus with original documents [3].

The Cross-Language Evaluation Forum, CLEF, provides an infrastructure for the evaluation of information retrieval systems, both monolingual and cross-lingual. We participated in the TEL@CLEF task of the CLEF 2009 ad-hoc track, which aims at the evaluation of systems to retrieve relevant items from multilingual collections of library catalog records. The main challenges of this task are the multilinguality and the sparsity of the dataset. We used two different CLIR approaches to tackle this task; the paper in hand outlines and discusses these approaches and the achieved results.

The first approach is Cross-Language Explicit Semantic Analysis, CL-ESA, which is a multilingual retrieval model to access cross-language similarity between text documents [3]. The CL-ESA model exploits a document-aligned comparable corpus such

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 50–57, 2010.

as Wikipedia in order to map the query and the documents into a common multilingual concept space [3,4]. We also present a formal definition and an alternative interpretation for the CL-ESA model, which is inspired by the Generalized Vector Space Model, GVSM. Our view is mathematically equivalent to the original idea of the CL-ESA model; it reveals how the computational effort for CL-ESA can be shifted from the query phase to a preprocessing phase.

In the second approach, called Cross Querying, each query is translated into all target languages. The particular rankings are used in a combined fashion considering the most likely language of the documents. The evaluation on the TEL@CLEF collections shows that both CLIR approaches are able to outperform the monolingual baseline. In the bilingual subtask, querying with a foreign language, Cross Querying achieves nearly the same or even higher results compared to the monolingual subtask; the performance of the CL-ESA is lower compared to the monolingual results.

The paper is organized as follows. Section 2 describes the target collection used in the TEL@CLEF task along with the evaluation procedure. Section 3 defines the general CL-ESA model, our formalization, and details of the CL-ESA implementation employed in the experiments. Section 4 presents the Cross Querying approach, Section 5 discusses the evaluation, and Section 6 concludes with an outlook.

2 TEL@CLEF Dataset and Evaluation Procedure

In this year's TEL@CLEF task three target collections, provided by The European Library, TEL, are used. The collections are labeled BL, ONB, and BNF, and mainly contain information in English, German, and French respectively (see Table 1). The collections are comprised of library catalog records, referring to different types of items such as articles, books, or videos. The data is provided in structured form and represented in XML. Each library catalog record has several fields containing meta information and content information that describe the particular item. Typical meta information fields are `author`, `rights`, or `publisher`, and typical content information fields are `title`, `description`, `subject`, or `alternative`. In our experiments we focus on the content information fields. A major difficulty is the sparsity of the available information: for many records only few fields are given.

The user's information need is specified by 50 topics that are provided by CLEF in the three main languages of the target collections, namely English, German, and French. A topic consists of two fields: a `title`, containing 2-4 keywords, and a `description`, containing 1-2 sentences that specify the item of interest in greater detail. The topics are used to construct the queries.

The TEL@CLEF task is divided into a monolingual and a bilingual subtask. The aim in both subtasks is to retrieve documents (library catalog records) from the target collections, which are most relevant to a query; for each query the results are submitted as a ranked list of documents. In the monolingual subtask the language of the query and the main language of the collection are the same, while in the bilingual subtask the language of the query is different from the main language of the collection. We submitted runs for both subtasks and for all three languages.

Table 1. Statistics of the three target collections used in the TEL@CLEF task: British Library, BL; Österreichische Nationalbibliothek, ONB; and Bibliothèque nationale de France, BNF

	BL	**ONB**	**BNF**
main language	English	German	French
# documents	1 000 100	869 353	1 000 100
# documents with `title`	1 000 042	829 675	1 000 095
average length of `title` per document	8.033	5.500	17.124
# documents with `description`	518 493	0	1 000 100
average length of `description` per document	6.222	0	10.095
# documents with `subject`	671 544	602 580	368 788
average length of `subject` per document	7.032	8.373	10.833
# documents with `alternative`	78 679	404 415	0
average length of `alternative` per document	5.491	8.158	0
# documents without content information	20	37 564	0

3 Cross-Language Explicit Semantic Analysis

Cross-Language Explicit Semantic Analysis, CL-ESA, is a generalization of the Explicit Semantic Analysis, ESA [2], and was proposed by Potthast et al. [3]. This section presents a formal definition of the CL-ESA model that reveals its close connection to the Generalized Vector Space Model, GVSM [5]: the ESA model and the GVSM can be transformed into each other [1]. It follows immediately that this is also true for the CL-ESA model and the cross-lingual extension of the Generalized Vector Space Model, CL-GVSM [6].

3.1 Formal Definition

Let d_i be a real-world document written in language L_i, and let $\mathbf{d}_i$ be a bag-of-word-based representation of d_i, encoded as a vector of normalized term frequency weights over a universal term vocabulary V_i. V_i contains all used terms for language L_i. A set $\mathbf{D}_i$ of document representations defines a term-document matrix A_{D_i}, where each column in A_{D_i} corresponds to a vector $\mathbf{d}_i \in \mathbf{D}_i$.

Definition 1 (ESA Representation [1]). *Let D_i^* be a collection of index documents written in language L_i. The ESA representation $\mathbf{d}_{iESA}$ of a document d_i with representation $\mathbf{d}_i$ is defined as follows:*

$$\mathbf{d}_{iESA} = A_{D_i^*}^T \cdot \mathbf{d}_i, \tag{1}$$

where A^T designates the matrix transpose of A.

The rationale of this definition becomes clear if one considers that the weight vectors $\mathbf{d}_i^* \in \mathbf{D}_i^*$ and $\mathbf{d}_i$ are normalized: $||\mathbf{d}_i^*|| = ||\mathbf{d}_i|| = 1$, for each $\mathbf{d}_i^* \in \mathbf{D}_i^*$. Hence, each entry in the ESA representation $\mathbf{d}_{iESA}$ of a document d_i is the cosine similarity between $\mathbf{d}_i$ and some vector $\mathbf{d}_i^* \in \mathbf{D}_i^*$. Put another way, d_i is compared to each index document in D_i^*, and $\mathbf{d}_{iESA}$ is comprised of the respective cosine similarities.

Definition 2 (CL-ESA Similarity). *Let $\mathcal{L} = \{L_1, \ldots, L_k\}$ denote a set of natural languages, and let $\mathcal{D}^* = \{D_1^*, \ldots, D_k^*\}$ be a set of index collections where each $D_i^* \in \mathcal{D}^*$ is a list of index documents written in language $L_i \in \mathcal{L}$. $\mathcal{D}^*$ is a document-aligned comparable corpus, i.e., for each language $L_i \in \mathcal{L}$ the n-th index document in $D_i^* \in \mathcal{D}^*$ describes the same concept. The CL-ESA similarity, $\varphi_{CL-ESA}(q_j, d_i)$, between a query q_j in language L_j and a document d_i in language L_i is computed as cosine similarity φ of the ESA representations of q_j and d_i:*

$$\varphi_{CL-ESA}(q_j, d_i) = \varphi(\mathbf{q}_{j\,ESA}, \mathbf{d}_{i\,ESA}) = \varphi(A_{D_j^*}^T \cdot \mathbf{q}_j, A_{D_i^*}^T \cdot \mathbf{d}_i) \tag{2}$$

Due to the alignment of the index collections D_j^* and D_i^* the ESA representations of q_j and d_i are comparable. Definition 2 is equivalent to the definition of the CL-GSVM similarity $\varphi_{CL-GVSM}(q_j, d_i)$ given in [6], which means that, in analogy to [1], the CL-ESA model and the CL-GVSM can be directly transformed into each other:

$$\varphi_{CL-ESA}(q_j, d_i) = \varphi(A_{D_j^*}^T \cdot \mathbf{q}_j, A_{D_i^*}^T \cdot \mathbf{d}_i) = \varphi_{CL-GVSM}(q_j, d_i) \tag{3}$$

3.2 Alternative Interpretation

The original idea of the CL-ESA model is to map both query and documents into a multilingual concept space, as it is expressed in Equation 2. Note that Equation 2 can be rearranged as follows:

$$\varphi_{CL-ESA}(q_j, d_i) = \varphi(A_{D_j^*}^T \cdot \mathbf{q}_j, A_{D_i^*}^T \cdot \mathbf{d}_i) = \mathbf{q}_j^T \cdot A_{D_j^*} \cdot A_{D_i^*}^T \cdot \mathbf{d}_i \tag{4}$$

In particular, the matrix $A_{D_j^*} \cdot A_{D_i^*}^T = G_{j,i}$ can be computed in advance since it is independent from a particular q_j or d_i. Hence:

$$\varphi_{CL-ESA}(q_j, d_i) = \mathbf{q}_j^T \cdot G_{j,i} \cdot \mathbf{d}_i \tag{5}$$

The rationale of Equation 5 becomes apparent if one recognizes $G_{j,i} = A_{D_j^*} \cdot A_{D_i^*}^T$ as $|V_j| \times |V_i|$ term co-occurrence matrix. The n-th row in $A_{D_j^*}$ corresponds to the distribution of the n-th term $t_n \in V_j$ over the index documents in D_j^*; likewise, the m-th row in $A_{D_i^*}$ corresponds to the distribution of the m-th term $t_m \in V_i$ over the index documents in D_i^*. Recall that the index documents in D_j^* and D_i^* are aligned. I.e., the value in the n-th row and the m-th column of $G_{j,i}$ quantifies the similarity between the distributions of t_j and t_i given the concepts described by the index documents in D_j^* and D_i^*.

The CL-ESA similarity computation of Equation 5 can be viewed in two ways:

(*i*) As a translation of the query representation $\mathbf{q}_j$ into the space of the document representation $\mathbf{d}_i$: $\varphi_{CL-ESA}(q_j, d_i) = (\mathbf{q}_j^T \cdot G_{j,i}) \cdot \mathbf{d}_i$, or,

(*ii*) as a translation of the document representation $\mathbf{d}_i$ into the space of the query representation $\mathbf{q}_j$: $\varphi_{CL-ESA}(q_j, d_i) = \mathbf{q}_j^T \cdot (G_{j,i} \cdot \mathbf{d}_i)$.

These views are different from the original idea of the CL-ESA model where both the query representation and the document representation are mapped into a common multilingual concept space (see Equation 2). From a mathematical standpoint Equation 2 and Equation 5 are equivalent; however, implementing CL-ESA based on the alternative

Table 2. The different interpretations of the CL-ESA model

	Original interpretation	**Alternative interpretation**	
		View (*i*)	View (*ii*)
$\varphi_{CL-ESA}(q_j, d_i) =$	$\varphi(A^T_{D^*_j} \cdot \mathbf{q}_j, A^T_{D^*_i} \cdot \mathbf{d}_i)$	$(\mathbf{q}_j^T \cdot G_{j,i}) \cdot \mathbf{d}_i$	$\mathbf{q}_j^T \cdot (G_{j,i} \cdot \mathbf{d}_i)$
Runtime complexity	$O(l \cdot \lvert D^* \rvert + \lvert D^* \rvert)$	$O(l \cdot \lvert V_j \rvert + l)$	$O(l)$

interpretation yields a considerable runtime improvement in practical retrieval applications. Table 2 contrasts the interpretations and the related runtime complexities. Here, we assume a closed retrieval situation where from a given target collection D_i in language L_i the most similar documents to a query q_j in language L_j are desired. CLIR with CL-ESA is straightforward: computation of $\varphi_{CL-ESA}(q_j, d_i)$ for each $d_i \in D_i$ and ranking by decreasing CL-ESA similarity.

Under the original interpretation the ESA representations $\mathbf{d}_{i\,ESA}$ of the documents $d_i \in D_i$ can be computed in advance. At retrieval time the query is mapped into the concept space in $O(l \cdot |D^*|)$, where l denotes the number of query terms. The computation of the cosine similarity between the ESA representations $\mathbf{q}_{j\,ESA}$ and $\mathbf{d}_{i\,ESA}$ requires $O(|D^*|)$. Under the alternative interpretation the matrix $G_{j,i}$ can be computed in advance. Note that in practical applications $l \ll |D^*|$, since a reasonable index collection size $|D^*|$ is 10 000, which shows the substantial performance improvement under the alternative interpretation and View (*ii*) .

3.3 Usage in TEL@CLEF

In this subsection we describe implementation details of the CL-ESA model we used in our submission. The following parameter setting was determined by analyzing unofficial experiments of the TEL@CLEF 2008 dataset.

Query and Document Construction. We use the original words of both topic fields, `title` and `description`, as queries. The documents are constructed by merging the text of the three record fields `title`, `subject`, and `alternative`. We assume that the language of these fields is the same within one record; however, this assumption may be violated in some cases since the collections contain multilingual records. Records containing non of these fields are omitted in the experiments (see Table 1).

Index Collection. As index collection Wikipedia is employed. We restrict the multilinguality of our model to the three main languages of the target collections: English, German, and French. Based on a Wikipedia snapshot from March 2009 about 169 000 articles per language can be aligned and fulfill several filter criteria, e.g., to contain more than 100 words or not to be a disambiguation or redirection page. All articles are used as index documents. As term weighting schema $tf \cdot idf$ is used. Query and document words are stemmed using the Snowball stemmers. To speed-up the CL-ESA similarity computation all values below a threshold of $\epsilon = 0.025$ are discarded.

Language Detection. While the language of the queries is determined by the corresponding topics the language of the documents is unknown since the collections are

multilingual and no language meta information is provided. In the experiments we resort to a simple "detection by stop words" approach that employs a stop word list for each of the three main languages and counts for each list the occurrences of the particular stop words within a document. A document is expected to have the language of the list with the highest count; if the detection is inconclusive the main language of the collection is assumed.

4 Cross Querying

Cross querying is a straightforward approach for CLIR systems. We subsume the fields of a topic in one query which is translated in the other languages. With each of the translations we compute a set of rankings by retrieving against each document field. The rankings are merged with respect to their cosine similarities. Additionally, the scores are multiplied by a boosting constant.

Definition 3 (Cross Querying). *Let $\mathcal{L} = \{L_1, \ldots, L_k\}$ denote a set of natural languages and let $\mathcal{F} = \{F_1, \ldots, F_k\}$ denote a set of document fields. $lang : D \rightarrow \mathcal{L}$, $lang(d) \mapsto L_i$ estimates the language of a document d. $\mathbf{d}, \mathbf{q}$, and $\mathbf{q}_{L_i}$ are the representations of a document d, a query q and the translation of q in language L_i. Then the cross querying similarity, $\varphi_{CQ}(q, d)$, of a query q and a document d is defined as follows:*

$$\varphi_{CQ}(q,d) = \sum_{F_i \in \mathcal{F}} \Big(b \cdot \varphi(\mathbf{q}_{lang(d)}, \mathbf{d}_{F_i}) \; + \sum_{\substack{L_i \in \mathcal{L}, \\ L_i \neq lang(d)}} \varphi(\mathbf{q}_{L_i}, \mathbf{d}_{F_i}) \Big), \tag{6}$$

where φ is the cosine similarity and b the boosting constant.

The name "Cross Querying" reflects the fact that $|\mathcal{L}| \times |\mathcal{F}|$ rankings are merged by querying in each language in each field. The applied parameters are as follows:

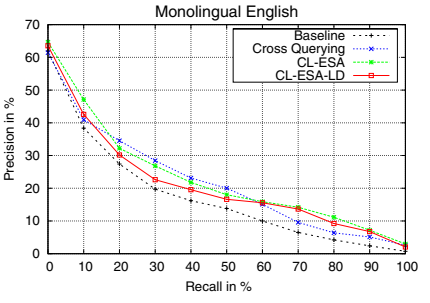
Query and Document Construction. The words of both topic fields, `title` and `description`, are used as queries and translated to each $L_i \in \mathcal{L}$, with $\mathcal{L} = \{German, French, English\}$. The selection of the document fields corresponds to `title` and `subject`. As term weighting schema $tf \cdot idf$ is used. Query and document words are stemmed using the Snowball stemmers while stop words are removed. The queries are translated with Google Translate; the boosting constant b is based on the unofficial evaluation on the TEL@CLEF 2008 dataset.

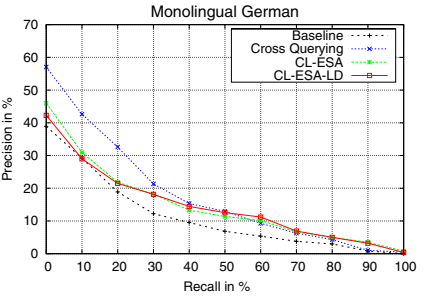
Language Detection. In order to estimate the language of d with $lang(d)$ we take the corpus language of the associated evaluation run.

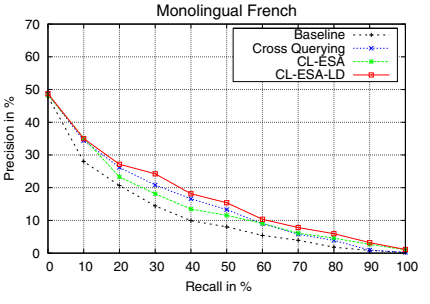
5 Evaluation Results

The results of the monolingual subtask and the bilingual subtask are shown in Figure 1 and Figure 2 respectively.

We submitted an additional baseline to the monolingual subtask using state-of-the-art retrieval technology: since in this subtask the language of the topics is equal to the main language of the target collection, the ranking is based on the cosine similarities of the $tf \cdot idf$-weighted bag-of-words representations of the topics and the documents.

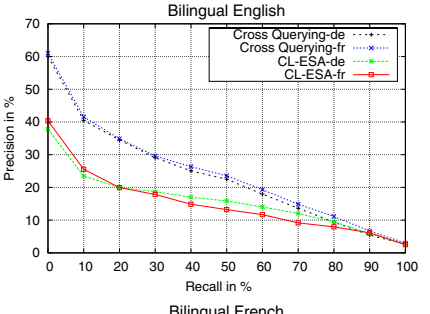


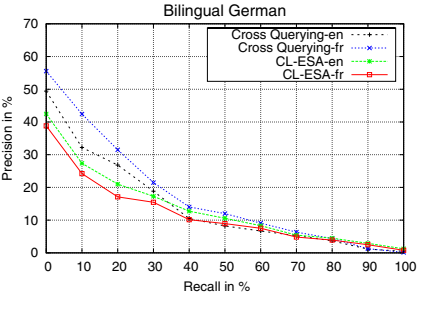


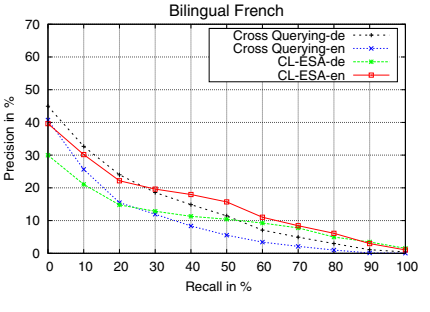


	English	German	French
Baseline	0.158	0.100	0.110
Cross Querying	0.200	**0.164**	0.145
CL-ESA	**0.215**	0.137	0.142
CL-ESA-LD	0.195	0.134	**0.163**

Fig. 1. Evaluation results of the monolingual runs. The plots show the standard recall levels vs. interpolated precision. The table show the results in terms of mean average precision, MAP.







	English	German	French
Cross Querying-en	-	0.129	0.132
Cross Querying-de	0.215	-	0.087
Cross Querying-fr	**0.225**	**0.158**	-
CL-ESA-en	-	0.124	**0.145**
CL-ESA-de	0.144	-	0.104
CL-ESA-fr	0.139	0.108	-

Fig. 2. Evaluation results of the bilingual runs. The plots show the standard recall levels vs. interpolated precision. The table show the results in terms of mean average precision, MAP.

Each plot in Figure 1 corresponds to one target collection and shows the baseline along with the results achieved under Cross Querying, CL-ESA, and CL-ESA with automatic language detection, CL-ESA-LD. Both Cross Querying and CL-ESA gain a higher MAP than the baseline. The variation between the two approaches is small, except for the German collection where Cross Querying outperforms CL-ESA at low recall levels. At higher recall levels CL-ESA is better, which explains a slightly higher MAP on the English and the French collections. Using CL-ESA along with the automatic language detection improves the performance only for the French collection, which indicates that this collection contains a larger fraction of non-French documents.

In the bilingual subtask the language of the queries is different from the main language of the target collection. Each plot in Figure 2 corresponds to one target collection that is queried in the two other languages, using both Cross Querying and CL-ESA. For example, in the plot "Bilingual English" the graph for "CL-ESA-de" shows the results of querying the English collection with German topics using the CL-ESA. Cross Querying achieves nearly the same or even higher results compared to the monolingual situation, whereas the CL-ESA performs worse in contrast to the monolingual results.

6 Conclusion and Future Work

The evaluation results for the TEL@CLEF task show that both CLIR approaches CL-ESA and Cross Querying are able to outperform the monolingual baseline—though the absolute results are still improvable. Furthermore, we have presented a formal definition and an alternative interpretation for the CL-ESA model, which is interesting for real-world retrieval applications since it reveals how the computational effort for CL-ESA can be shifted from the query phase to a preprocessing phase.

As for future work, CL-ESA and Cross Querying will benefit if more languages are taken into account. Currently, German, English, and French are used, but the target collections comprise more languages. For documents from other languages an inconsistent CL-ESA representation is computed. CL-ESA also needs a reliable language detection mechanism in order to compute a consistent representation; note that we used a rather simple approach in our experiments.

References

1. Anderka, M., Stein, B.: The ESA Retrieval Model Revisited. In: Proc. of SIGIR 2009, pp. 670–671 (2009)
2. Gabrilovich, E., Markovitch, S.: Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In: Proc. of IJCAI 2007, pp. 1606–1611 (2007)
3. Potthast, M., Stein, B., Anderka, M.: A Wikipedia-Based Multilingual Retrieval Model. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 522–530. Springer, Heidelberg (2008)
4. Sorg, P., Cimiano, P.: Cross-lingual Information Retrieval with Explicit Semantic Analysis. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 243–250. Springer, Heidelberg (2009)
5. Wong, S.K.M., Ziarko, W., Wong, P.C.N.: Generalized Vector Spaces Model in Information Retrieval. In: Proc. of SIGIR 1985, pp. 18–25 (1985)
6. Yang, Y., Carbonell, J.G., Brown, R.D., Frederking, R.E.: Translingual Information Retrieval: Learning from Bilingual Corpora. Artif. Intell. 103(1-2), 323–345 (1998)

Document Expansion, Query Translation and Language Modeling for Ad-Hoc IR

Johannes Leveling[1], Dong Zhou[2], Gareth J.F. Jones[1], and Vincent Wade[2]

[1] Centre for Next Generation Localisation
School of Computing
Dublin City University, Dublin 9, Ireland
{johannes.leveling,gareth.jones}@computing.dcu.ie
[2] Centre for Next Generation Localisation
Computer Science Department
Trinity College Dublin, Dublin, Ireland
{dong.zhou,vincent.wade}@cs.tcd.ie

Abstract. For the multilingual ad-hoc document retrieval track (TEL) at CLEF, Trinity College Dublin and Dublin City University participated in collaboration. Our retrieval experiments focused on i) document expansion using an entry vocabulary module, ii) query translation with Google translate and a statistical MT system, and iii) a comparison of the retrieval models BM25 and language modeling (LM). The major results are that document expansion did not increase MAP; topic translation using the statistical MT system resulted in about 70% of the mean average precision (MAP) achieved compared to Google translate, and LM performs equally or slightly better than BM25. The bilingual retrieval French and German to English experiments obtained 89% and 90% of the best MAP for monolingual English.

1 Introduction

The TEL (The European Library) task at CLEF is concerned with ad-hoc information retrieval (IR) [1]. Our IR experiments for the ad-hoc IR task at CLEF 2009 aim at investigating several aspects of retrieval: evaluating document expansion (DE) to obtain longer documents for the TEL collection; applying statistical MT [2] for topic translation and comparing it to Google translate, and comparing retrieval by language modeling (LM) [3] with Okapi BM25 [4].

2 Retrieval Experiments

The Lemur toolkit[1] was employed to index and retrieve documents. Two different retrieval models were used: BM25 [4] with default parameters ($b = 1.2$, $k1 = 2.0$, $k3 = 7$) and LM with Jelinek-Mercer smoothing [3]. TEL documents follow the Dublin Core metadata standard and contain multiple fields including title,

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

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 58–61, 2010.

contributors, language, and subject terms. For different experiments, the text of different document fields was extracted and processed to produce a single flat index. Prior to indexing the documents, their contents were preprocessed with the Snowball stemmer[2] and stopwords were removed. (see [5] for a more detailed description of indexed fields and document preprocessing).

For most runs, pseudo-relevance feedback was applied for query expansion (QE): the top ten ranked documents and 30 terms were used for BM25 and the top five documents and 20 added terms for LM. A variant of query expansion using information from an external resource was also explored for bilingual retrieval (QE2). The top 10 results for the query in the source language were identified and translated with Google translate. Highly co-occurring terms were extracted for query expansion, using the mutual information to calculate co-occurrence and select the highest score for target translation. For the bilingual retrieval experiments, topics were translated using either Google translate (GT)[3] or a statistical machine translation system (MT) [2].

3 Document Preprocessing

The main idea for document expansion was to train a classifier on documents containing a Dewey Decimal Code (DDC) to obtain classification codes for all documents. All classification codes are then replaced with their natural language description, which is added to the document before indexing. The natural language descriptions are available in English only and originate from the OCLC web site[4]. The complete natural languages descriptions for DDC contain 1110 entries of which 933 were actually used in the document collection.

We trained an EVM (Entry Vocabulary Module, [6]) on all documents containing a DDC and applied it to select the top-ranked DDC. Documents with a DDC are expanded before indexing by replacing the code with its natural language description; documents without a DDC are first classified using the EVM and then processed as described above.

4 Results and Conclusions

Results for the ad-hoc IR experiments are shown in Table 1. Some experiments achieved a performance among the top five participants at the TEL track at CLEF 2009, i.e. run DEEN1[5] was 4th in bilingual English (0.3333 MAP), run DE3 was 4th in monolingual German (0.2686 MAP), and run EN3 was 5th in monolingual English (0.3696 MAP).

In all cases, runs with blind relevance feedback to expand queries yield a higher MAP compared to the corresponding runs without blind feedback. The

[2] http://snowball.tartarus.org/
[3] http://translate.google.com/
[4] http://www.oclc.org/dewey/
[5] The prefix TCDDCU has been omitted from the run labels for brevity.

query expansion variant based on external information from web pages found by Google web search did not show the expected results as it degraded the performance (DEEN3 vs. DEEN1).

For the bilingual runs with target language English, 89.9% and 90.1% of the MAP for the best monolingual English runs was achieved for French and German, respectively. Using the MaTrEx system for topic translation achieves a MAP of 70.1% in comparison to topic translation by Google translate (FREN2 vs. FREN1).

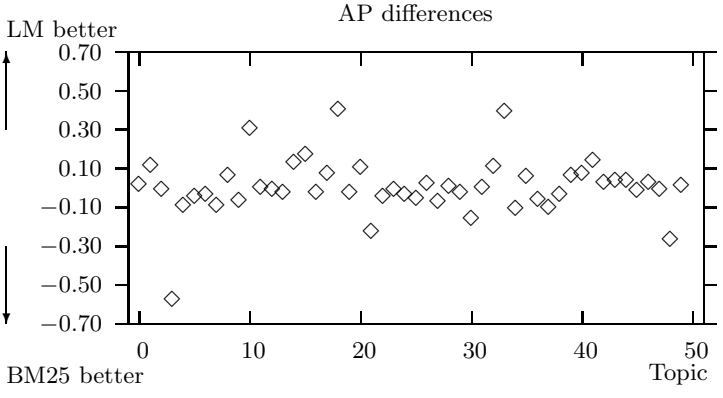


Fig. 1. Differences in AP for English BM25 and LM experiments

To investigate differences in results for the retrieval models BM25 and LM for monolingual IR, we compared the average precision (AP) for the best runs in English, French, and German (run EN3 vs. EN1F, DE3 vs. DE1F, and FR3 vs. FR1F). A comparison of the English runs EN3 and EN1F is shown in Figure 1. While there seem to be only small changes in performance for the different languages and retrieval models, there is also a small number of topics for each language where the IR models seem to behave very differently. For example for topic 12, LM yields a higher AP compared to BM25 for French; for German, the opposite effect can be observed for this topic. In computing the AP differences, we found that LM returns a higher AP than BM25 for 23 English topics, a lower AP for 26 topics, and the same AP for one topic. For French (German), LM yields a higher AP than BM25 for 29 (23) topics and a lower AP for 21 (27) topics. On average, LM improved precision of slightly less topics compared to BM25, but it resulted in a higher MAP. In conclusion, these IR models seem to return results with similar AP values, but can also behave very differently for certain topics. Further research is required to determine if the best retrieval model for a topic in a given language can be selected automatically or how retrieval results can best be combined.

Table 1. Results for monolingual and bilingual IR experiments for the ad-hoc task

Run ID	source	target	description	MAP	GMAP	P@10
EN1F	EN	EN	BM25, subset, QE	0.3640	0.1926	0.5080
EN2F	EN	EN	BM25, subset, QE, DE	0.3426	0.1869	0.4980
EN3	EN	EN	LM, subset, QE	0.3696	0.2414	0.5060
EN4	EN	EN	LM, all, QE	0.3688	0.2675	0.5200
FR1	FR	FR	BM25, subset	0.1783	0.0982	0.3340
FR1F	FR	FR	BM25, subset, QE	0.1831	0.0919	0.3420
FR3	FR	FR	LM, subset, QE	0.1758	0.0434	0.2327
FR4	FR	FR	LM, all, QE	0.1749	0.0417	0.2224
DE1	DE	DE	BM25, subset	0.2329	0.1221	0.3540
DE1F	DE	DE	BM25, subset, QE	0.2561	0.1137	0.3580
DE3	DE	DE	LM, subset, QE	0.2686	0.1291	0.3840
DE4	DE	DE	LM, all, QE	0.2439	0.1258	0.3460
DEEN1	DE	EN	LM, GT, subset, QE	0.3333	0.1981	0.4420
DEEN3	DE	EN	LM, GT+QE, subset, QE2	0.2947	0.1351	0.3900
FREN1F	FR	EN	BM25, GT, subset, QE	0.3323	0.1761	0.4820
FREN2	FR	EN	BM25, MT subset,	0.2072	0.0533	0.3800
FREN2F	FR	EN	BM25, MT, subset, QE	0.2551	0.0497	0.3920

Acknowledgments

This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/CE/I1142.

References

1. Agirre, E., Di Nunzio, G.M., Ferro, N., Mandl, T., Peters, C.: CLEF 2008: Ad hoc track overview. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 15–37. Springer, Heidelberg (2009)
2. Du, J., He, Y., Penkale, S., Way, A.: MaTrEx: the DCU MT system for WMT 2009. In: Proceedings of the Fourth Workshop on Statistical Machine Translation, Athens, Greece, pp. 95–99 (2009)
3. Brown, P.F., Cocke, J., Della Pietra, S.A., Della Pietra, V.J., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Computational Linguistics 16(2), 79–85 (1990)
4. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.: Okapi at TREC-3. In: Proceedings of the Third Text REtrieval Conference, Gaithersburg, USA (1994)
5. Leveling, J., Zhou, D., Jones, G., Wade, V.: TCD-DCU at TEL@CLEF 2009: Document expansion, query translation, and language modeling. In: Working Notes of the CLEF 2009 Workshop, Corfu, Greece, September 30 -October 2 (2009)
6. Gey, F.C., Buckland, M., Chen, A., Larson, R.R.: Entry vocabulary – a technology to enhance digital search. In: Proceedings of the First International Conference on Human Language Technology, San Diego, USA (2001)

Smoothing Methods and Cross-Language Document Re-ranking

Dong Zhou and Vincent Wade

Centre for Next Generation Localisation
Knowledge and Data Engineering Group
Trinity College Dublin
Dublin 2, Ireland
dongzhou1979@hotmail.com , vincent.wade@cs.tcd.ie

Abstract. This paper presents a report on our participation in the CLEF 2009 monolingual and bilingual *ad hoc* TEL@CLEF task involving three different languages: English, French and German. Language modeling was adopted as the underlying information retrieval model. While the data collection is extremely sparse, smoothing is particularly important when estimating a language model. The main purpose of the monolingual tasks is to compare different smoothing strategies and investigate the effectiveness of each alternative. This retrieval model was then used alongside a document re-ranking method based on Latent Dirichlet Allocation (LDA) which exploits the implicit structure of the documents with respect to original queries for the monolingual and bilingual tasks. Experimental results demonstrated that three smoothing strategies behave differently across testing languages while the LDA-based document re-ranking method should be considered further in order to bring significant improvement over the baseline language modeling systems in the cross-language setting.

1 Introduction

This year's participation in the CLEF 2009 *ad hoc* monolingual and bilingual track was motivated by a desire to compare different smoothing strategies applied to language modeling for library data retrieval as well as to test and extend a newly developed document re-ranking method.

Language modeling has been successfully applied to the problem of ad hoc retrieval [1,3]. It provides an attractive information model due to its theoretical foundations. The basic idea behind this approach is extremely simple - estimate a language model for each document and/or a query, and rank documents by the likelihood of the query (with respect to the document language model) or by the distance between the two models. The main object of smoothing is to adjust the maximum likelihood estimator of a language model so that it will be more accurate [3].

However, previous success over news collection data does not necessarily mean it will be efficient over the library data. Firstly the data is actually multilingual:

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 62–69, 2010.

all collections to a greater or lesser extent contain records pointing to documents in other languages. However this is not a major problem because the majority of documents in the test collection are written in the main languages of those test collections. Furthermore, the main characteristic of the data is that it is very different from the newspaper articles and news agency dispatches previously used in the CLEF. The data tends to be very sparse. Many records contain only title, author and subject heading information; other records provide more detail (see the experiment section on what fields are chosen for inclusion). The average document lengths are 14.66 for British Library (BL) and 24.19 for Bibliothèque nationale de France (BNF) collections after pre-processing, respectively.

A more recent trend is to explore the hidden structure of documents to re-rank results [4]. We claimed in a previous work [4] that there are two important factors that should be taken into account when designing any re-ranking algorithm: the original queries and the initial retrieval scores. Based on this observation, we introduce a new document re-ranking method based on Latent Dirichlet Allocation (LDA) which exploits implicit structure of the documents with respect to original queries. Rather than relying on graph-based techniques as in [5, 6] to identify the internal structure, the approach tries to directly model the latent structure of "topics" or "concepts" in the initial retrieval set. Then we can compute the distance between queries and initial retrieval results based on latent semantic information inferred. Experiments in [4] demonstrated the effectiveness of the proposed method in monolingual retrieval. In this experiment, we try to extend the approach to cross-language information retrieval.

2 Methodology

2.1 Language Modeling

Smoothing a data set typically means creating an approximating function that attempts to capture important patterns in the data, while leaving out noise or other fine-scale structures/rapid phenomena. In language modeling, the basic reason to use smoothing is to ensure we do not assign a zero probability to unseen words. The accuracy of smoothing is directly related to the retrieval performance.

Given a text sequence (either a query or a document), the probability distribution can be regarded as a probabilistic language model M_d or M_q from each document d or each query q. In other words, it assumes that there is an underlying language model which "generates" a term (sequence) [1]. The unigram language model is utilized here. There are several ways to estimate the probabilities. Let $g(w \in d)$ denotes the number of times the term w occurs in a document d (same idea can be used on a query). The Maximum-likelihood estimation (MLE) of w with respect to d is defined as:

$$MLE_d w = \frac{g(w \in d)}{\sum_{w'} g(w' \in d)} \tag{1}$$

We choose to use three representative methods that are widely used in previous research and relatively efficient to implement. The first method we adopt is the Jelinek-Mercer method, defined as:

$$JM_d w = (1 - \lambda) \cdot MLE_d w + \lambda \cdot MLE_{\mathbf{D}} w \tag{2}$$

where smoothing parameter λ (same as μ and δ used in the following methods) controls the degree of reliance on relative frequencies in the document corpus rather than on the counts in d. The second method used is called Bayesian smoothing using Dirichlet prior:

$$DIR_d w = \frac{g(w \in d) + \mu \cdot MLE_{\mathbf{D}} w}{\sum_{w'} g(w' \in d) + \mu} \tag{3}$$

and the third method is the absolute discounting, defined as:

$$ABS_d w = \frac{max(g(w \in d) - \delta, 0)}{\sum_{w'} g(w' \in d)} + \delta \cdot \frac{|d|_\mu}{|d|} \cdot MLE_{\mathbf{D}} w \tag{4}$$

where $|d|_\mu$ is the number of unique terms in document d and $|d|$ is the total count of words in the document. Note that $|d| = \sum_{w'} g(w' \in d)$. This concludes our description of the smoothing methods employed in the experiments.

2.2 Document Re-ranking

The intuition behind the document re-ranking method is the hidden structural information among the documents: *similar documents are likely to have the same hidden information with respect to a query.* In other words, if a group of documents are talking about the same topic which shares a strong similarity with a query, in our method they will get allocated similar ranking as they are more likely to be relevant to the query. In addition, the refined ranking scores should be relevant to the initial ranking scores, which, in the experiments conducted in this paper, are combined together with the re-ranking score using a linear fashion.

The distance between a query and a document in this method adopts the KL divergence between the query terms and document terms to compute a Re-Rank score RS^{KL1}_{LDA}:

$$RS^{KL1}_{LDA} = -D(MLE_q(\cdot)||LDA_d(\cdot)) \tag{5}$$

where

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{6}$$

The LDA based generative model is defined as:

$$LDA_d w = \sum_{z=1}^{k} p(w|z)p(z|d) \tag{7}$$

Then we formulate our method through a linear combination of the re-ranking scores based on initial ranker and the latent document re-ranker, shown as follow:

$$RS = (1 - \alpha) \cdot OS + \alpha \cdot RS_{LDA}^{KL1} \tag{8}$$

where OS denotes original scores returned by the initial ranker and α is a parameter that can be tuned with $\alpha = 0$ meaning no re-ranking is performed.

This method can be found in greater detail in [4]. We apply this method to the cross-language re-ranking by concatenating texts from different languages into several dual-language documents and a single dual-language query. An LDA analysis of these texts results in a multilingual semantic space in which terms from both languages are presented. Henceforth the re-ranking process can be carried out by directly modeling the latent structure of multilingual "topics" or "concepts" in this enriched initial retrieval set. The similarity of "contexts" in which the terms appear is guaranteed to capture the inter-relationship between texts in different languages.

3 Experimental Setup

3.1 Overview of the Experimental Process

All of the documents in the experiment were indexed using the Lemur toolkit[1]. Prior to indexing, Porter's stemmer and a stopword list[2] were used for the English documents. We use a French analyzer[3] and a German analyzer to analyze French and German documents. The query sets consist of 50 topics, all of which were used in the experiment. Each topic is composed of several parts: *Title*, *Description*, *Narrative*. We chose to use *Title+Description* fields to construct our queries. The queries are processed similarly to the treatment in the test collections (linguistic parsing). The chosen fields used in the indexing and searching stages are shown in the Table 1. Four metrics are adopted in the evaluation of the tasks: namely MAP, bref, P@5 and P@10 (top n results are particularly important).

3.2 Experimental Runs

In order to investigate the effectiveness of various techniques, we performed a retrieval experiment with several permutations. These experimental runs are denoted as follows:

For monolingual retrieval
LM-DIR: This part of the experiment involved retrieving documents from the test collection using language modeling with Bayesian smoothing method using Dirichlet prior.

[1] http://www.lemurproject.org
[2] ftp://ftp.cs.cornell.edu/pub/smart/
[3] http://lucene.apache.org/

Table 1. Indexing and searching fields

Fields	English	French	German
dc:language	√	√	√
dc:identifier	√	√	√
dc:rights	√	√	√
dc:type	√	√	√
dc:creator	√	√	√
dc:publisher	√	√	√
dc:date	√	√	√
dc:relation	√		
dc:contributor	√	√	√
dcterms:issued	√		√
dcterms:extent	√		
dcterms:spatial			√
dcterms:isPartOf			√
dcterms:edition			√
dcterms:available			√
mods:location	√	√	√

LM-ABS: as above, except that the absolute discounting smoothing method was used.

LM-JM: as above, except that the Jelinek-Mercer smoothing method was adopted.

For bilingual retrieval

GOOGLETRANS: In this part of the experiment, documents were retrieved from the test collection using the Google Translator[4] for translating the queries. (It is worth noting that due to the submission restrictions this is an unofficial experiment.)

GOOGLETRANS-LDA: Here we retrieved documents from the document collection using query translations suggested by the Google Translator. Then we directly re-rank the retrieval results using the translated query with the proposed LDA based document re-ranking method.

GOOGLETRANS-SLDA: Here we retrieved documents from the document collection using query translations suggested by the Google Translator. Then we built a multilingual corpus with documents written in both query and document languages. Re-ranking was performed by applying the LDA based method on this multilingual space (with the translated and the original query).

4 Results and Discussion

4.1 Monolingual Task

In this section we compare three smoothing methods across different languages in the library search (Table 2). As we conducted queries using the title and

[4] http://translate.google.com/

description fields, they could be considered as long informative queries. Previous research on news and web data [3] suggested that on average, Jelinek-Mercer is better than Dirichlet and absolute discounting for metrics such as non-interpolated average precision, precision at 10 and 20 documents. Also both Jelinek-Mercer and Dirichlet clearly have a better average precision than absolute discounting. The German monolingual runs demonstrated the same observation indicating that Jelnek-Mercer is better than Dirichlet, while Dirichlet is in turn better than absolute discounting.

The English and French runs showed a different behaviour. Absolute discounting was a clear winner among the three smoothing methods, whereas Jelinek-Mercer still performed better than Dirichlet. This may be explained by two different roles in the query likelihood retrieval method [3]. Usually the Dirichlet method performs better with shorter queries (estimation role). However in the experiments described in this paper only long queries were used. So that Dirichlet consistently demonstrated the worst performance across all the languages. However, Jelinek-Mercer performed best for longer queries and should be good for the role of query modeling. This was the case for the German runs while it was not the case for the English and French runs, in which absolute discounting substituted the Jelinek-Mercer's role in the modeling process. The results suggest that smoothing methods tend to be sensitive for distinct languages and different test collections.

Table 2. Retrieval results for monolingual task

Run ID	source	target	description	MAP	bpref	P@5	P@10
TCDENRUN1	EN	EN	LM-DIR	0.2905	0.3001	0.4560	0.4140
TCDENRUN2	EN	EN	LM-ABS	0.4035	0.4054	0.6160	0.5640
TCDENRUN3	EN	EN	LM-JM	0.3696	0.3658	0.5680	0.5060
TCDFRRUN1	FR	FR	LM-DIR	0.1451	0.1570	0.2000	0.1740
TCDFRRUN2	FR	FR	LM-ABS	0.1745	0.1767	0.2320	0.2380
TCDFRRUN3	FR	FR	LM-JM	0.1723	0.1765	0.2520	0.2280
TCDDERUN1	DE	DE	LM-DIR	0.2577	0.2615	0.4480	0.3760
TCDDERUN2	DE	DE	LM-ABS	0.2397	0.2397	0.4280	0.3540
TCDDERUN3	DE	DE	LM-JM	0.2686	0.2653	0.4520	0.3840

4.2 Bilingual Task

We now consider the bilingual tasks in order to study the LDA-based re-ranking method. The main experimental results are presented in Table 3, for all three languages. The first question we were interested in was how the re-ranking method performs directly over the bilingual retrieval results (taken as a whole). It is shown that our methods bring improvements upon the Google translator baselines in all of the 6 relevant comparisons. Another observation was that in many cases, the method can outperform the baselines for all the evaluation metrics.

Table 3. Retrieval results for bilingual task

Run ID	source	target	description	MAP	bpref	P@5	P@10
TCDFRENRUN1	FR	EN	GOOGLETRANS	0.3481	0.3526	0.5760	0.5220
TCDFRENRUN2	FR	EN	GOOGLETRANS-LDA	0.3488	0.3527	0.5720	0.5220
TCDFRENRUN3	FR	EN	GOOGLETRANS-SLDA	0.3500	0.3535	0.5760	0.5140
TCDDEENRUN1	DE	EN	GOOGLETRANS	0.3411	0.3500	0.5700	0.5040
TCDDEENRUN2	DE	EN	GOOGLETRANS-LDA	0.3500	0.3596	0.5760	0.5040
TCDDEENRUN3	DE	EN	GOOGLETRANS-SLDA	**0.3505**	0.3602	0.5880	0.5040
TCDENFRRUN1	EN	FR	GOOGLETRANS	0.1579	0.1572	0.2520	0.2320
TCDENFRRUN2	EN	FR	GOOGLETRANS-LDA	0.1591	0.1573	0.2520	0.2340
TCDENFRRUN3	EN	FR	GOOGLETRANS-SLDA	0.1576	0.1561	0.2560	0.2320
TCDDEFRRUN1	DE	FR	GOOGLETRANS	0.1618	0.1743	0.2680	0.2300
TCDDEFRRUN2	DE	FR	GOOGLETRANS-LDA	**0.1633**	0.1752	0.2600	0.2300
TCDDEFRRUN3	DE	FR	GOOGLETRANS-SLDA	0.1624	0.1739	0.2600	0.2260
TCDENDERUN1	EN	DE	GOOGLETRANS	0.1901	0.1923	0.3480	0.2900
TCDENDERUN2	EN	DE	GOOGLETRANS-LDA	0.1910	0.1922	0.3480	0.2920
TCDENDERUN3	EN	DE	GOOGLETRANS-SLDA	**0.1935**	0.1944	0.3480	0.2920
TCDFRDERUN1	FR	DE	GOOGLETRANS	0.1826	0.2053	0.3480	0.2700
TCDFRDERUN2	FR	DE	GOOGLETRANS-LDA	0.1840	0.2063	0.3520	0.2780
TCDFRDERUN3	FR	DE	GOOGLETRANS-SLDA	0.1839	0.2050	0.3560	0.2760

With respect to the bilingual re-ranking, the method showed some improvements over the Googe translator and direct re-ranking methods in the X2EN and X2DE runs in terms of mean average precision. The performance was somewhat disappointing in the X2FR runs. Furthermore, the improvements were not large enough in MAP. However, in terms of traditional re-ranking measurements such as precision at 5 documents, the method could demonstrate a higher performance than simple re-ranking. This showed that the method is a promising direction but further investigation will be needed.

It is worth mentioning that the combination of methods used in this experiment could achieve a very good overall performance as nearly all of our selected monolingual and bilingual runs were among top five participants in CLEF 2009 (except in the French monolingual task) such as:

TCDENRUN2 absolute discounting, English monolingual
TCDDERUN1 Dirichlet prior, German monolingual
TCDDEENRUN3 Google translator with SLDA, German-English bilingual
TCDDEFRRUN2 Google translator with LDA, German-French bilingual
TCDENDERUN3 Google translator with SLDA, English-German bilingual

5 Conclusion

In this paper we have described our contribution to the CLEF 2009 *ad hoc* monolingual and bilingual tracks. Our monolingual experiment involved the comparison of three different smoothing strategies applied to a language modeling approach for library data retrieval. We also made a first attempt to extend the previously proposed document re-ranking method to cross-language information retrieval. Experimental results demonstrated that smoothing methods tend to behave differently in the library search and across testing languages. They also showed that LDA-based document re-ranking method should be considered further in order to bring significant improvement over the baseline language modeling systems in the cross-language setting.

Acknowledgments. This research is supported by the Science Foundation of Ireland (grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College Dublin.

References

1. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and development in Information Retrieval, Melbourne, Australia, pp. 275–281. ACM, New York (1998)
2. Wei, X., Croft, W.B.: Lda-based document models for ad-hoc retrieval. In: Proceedings of the 29th Annual International ACM SIGIR conference on Research and Development in Information retrieval, Seattle, Washington, USA, pp. 178–185. ACM, New York (2006)
3. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (2004)
4. Zhou, D., Wade, V.: Latent document re-ranking. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, Singapore, pp. 1571–1580. Association for Computational Linguistics (2009)

Cross-Language Information Retrieval Using Meta-language Index Construction and Structural Queries

Amir Hossein Jadidinejad and Fariborz Mahmoudi

Electrical and Computer Engineering Department,
Islamic Azad University, Qazvin Branch
Qazvin, Iran
amir@jadidi.info, mahmoudi@qiau.ac.ir

Abstract. Structural Query Language allows expert users to richly represent its information needs but unfortunately, the complexity of SQLs make them impractical in the Web search engines. Automatically detecting the *concepts* in an unstructured user's information need and generating a richly structured, multilingual equivalent query is an ideal solution. We utilize Wikipedia as a great concept repository and also some state of the art algorithms for extracting Wikipedia's concepts from the user's information need. This process is called *"Query Wikification"*. Our experiments on the *TEL* corpus at *CLEF2009* achieves *+23%* and *+17%* improvement in Mean Average Precision and Recall against the baseline. Our approach is unique in that, it does improve both precision and recall; two pans that often improving one, hurt the another.

1 Introduction and Motivation

Representing user's information need is a fundamental part in an information retrieval system. Most systems get a list of keywords for each information need. For example, if a user is interested in "colour therapy and the therapeutic use of colour" they might formulate the natural language query *"colour therapy"*. It's not only a hard task for ordinary users to represent their information needs as a set of keywords but also clear that some semantic aspects are lost by transcribing the information need into a set of keywords. Such a query may retrieve some documents about "color" or "therapy" that completely irrelevant. Also, these models are incapable to address the two main problems of Natural Language Processing, *synonymy* and *polysemy*.

Combining the language model and inference network, as implemented in the *Indri* search engine, is efficient and verified approach. In this retrieval model, the user's information need is exhibited as *Indri's Structural Query Language*. Structural Query Languages (SQL) explicitly represents user's information needs and allows terms weighting, defining synonyms, the use of proximity information among terms, field restricting and various ways of combining concepts [7]. Since structured queries can be more expressive than keywords, it's verified that

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 70–77, 2010.

structural retrieval models such as Indri [13] and InQuery [2] have more potential to retrieve more accurate results. Although the structured queries and related models achieved a very good results in different experiments and applications but they suffer from a drawback that made them unusable in the Web search engines. Having knowledge about related concepts in the query is necessary to constructing structured queries. Even if we presume that the user has a good knowledge about its information need, learning the complicated Structured Query Languages for Web users is not desirable. Understanding the user's information need and generating a richly structured query can be an ideal solution. It needs a comprehensive concept repository that covers all query's concepts and a way to extracting appropriate concepts from the user's information need. The importance of both has been emphasized in previous researches [9] [10].
For example, take a look at the following user's information need[1]:

```
<topic lang="en">
     <identifier>10.2452/702-AH</identifier>
     <title>Colour Therapy</title>
     <description>
         Find books on the therapeutic use of colour.
     </description>
     <narrative/>
</topic>
```

After removing redundant and stop words, this information need can be represented by the following query:

```
colour therapy therapeutic
```

This query is not as informative as the original one because *words* are not good features to representing a fragment of text but *concepts* are ideal. Word-based information retrieval models (sometimes called "Bag Of Words" models) are as simple as a pattern matching system that matches the occurrences of the query words in the documents. It's clear that such systems are incapable to address the complexity of human languages such as synonymy and polysemy. The limitations of such models are more sensible in cross-language environments when dealing with retrieving information written in a language different from the language of the user's query. *Is it possible to extract a list of concepts in the information need and generate a rich structured query?*

To accomplish this idea, We need a *concept extractor* that elicits most important concepts from the user's information need. Suppose that the most important concepts of this query are extracted and the original query is annotated[2]:

```
[Chromotherapy | Colour Therapy]
Find books on the [Therapy | therapeutic] use of [Color | colour].
```

[1] Topic No. 10.2452/702AH at CLEF2009.
[2] Each concept is represented in the brackets. Details will be discussed in Sec. 3.

Note that these are *concepts* not *words*. It means that at least we have a list of synonyms, related terms and different translations for each concept. Using these concepts, we can automatically construct an Indri structured query [13] as follow:

```
#combine(colour therapy therapeutic
  #syn(chromotherapy farbtherapi colourology #1(color therapy))
  #syn(color couleur farb colour colors colours couleur)
  #syn(therapi thrap therapi treatment therapie therapy))
```

This structured query is more expressive. Some technical terms such as "chromotherapy" and "colourology" are included in the structured query that couldn't achieved from the original information need. Also, each concept has equivalents in different languages such as "farbtherapi". This query is achieved *+65%* improvements in Mean Average Precision while retrieving more relevant documents in our experiments.

In the following sections we elaborate on primary components of the proposed system. Sec. 2 addresses our indexing approach, Sec. 3 introduces the concept extractor algorithm, Sec. 4 explains our different strategies for automatically structured query construction and finally in Sec. 5, our system is evaluated and compared with others. The contributions of this paper are the following:

- Propose a new method for transcribing a well-formed, rich, efficient, structured query from a simple natural language information need. This process is done with the aid of state of the art algorithms in both "Wikification" [9,10] and "Structural Retrieval Models" [7]. It can substitute keyword-based Web search engines with the powerful structured queries and more efficient retrieval models.
- Evaluating the proposed approach in Cross-Language Information Retrieval. It made a meta-language search engine from Indri [13] that can efficiently apply on multilingual environments. Also the proposed method have a good potential to apply on the Web search engines in future.

2 Meta-language Index Construction

The TEL corpus used in the CLEF2009 Ad Hoc track is an inherently multilingual corpus that has been derived from a collection of The European Library [3]. It contains not only records in different languages but also some records may have multilingual fields. Previous experiments in the last year, utilize different language identification approaches to detect each field's language and then apply appropriate stemmer and stop words but lead to poor results [1]. We utilize a *meta-language index* in our experiments. Instead of distinguishing different languages, all informative fields are indexed without any concern about underlying language, stemming and stop word removal. It is clear that such indexing strategy is not appropriate in general but our experiments have shown that it is an appropriate indexing strategy in tandem with *Query Wikification* and *Indri Structured Query Language*. In the preprocessing step, we delete all noisy

and invaluable fields from the TEL corpus. After manual analysis of the TEL's records, we extract a list of fields that contains informative texts. Table 1 shows the valuable fields in preprocessing step. For more information about preprocessing and pruning of the dataset please refer to [5]. We utilize Indri [13] Field Index as our indexing engine because it not only construct a powerful field index but also support index's fields in its query language. Finally, the indexing is done using the Lemur toolkit [12].

Table 1. Valuable fields in the preprocessing step [5]

Title	Distribution	Description
dc:title	80%	This is record's title. All records contains this field and it is a valuable field.
dcterms:alternative	little	In some records, this field contains relevant information.
dc:subject	210%	Manually assigned subject heading.
dc:abstract	little	Record's abstract.
dc:description	42%	Record's description. Mostly contains copyrights and related stuffs.
dc:contributor	little	Record's contributor.

3 Query Wikification

The process of automatically recognizing the topics mentioned in an unstructured text and linking them to the appropriate Wikipedia [6] articles is known as *wikification* [9]. Two Wikification method have been proposed by now. The first is Wikify! [9] and the second is WM-Wikifier [10][3]. WM-Wikifier is a distinguish approach that uses Wikipedia articles not only as a source of information to point to, but also as training data for how best to create links [10]. It freely availables in the Wikipedia-Miner toolkit [11].

The user's information need is a short and informative text. So we can apply wikification algorithm on user's information needs in order to map unstructured query into a weighted list of concepts in the Wikipedia. We call this process as *"Query Wikification"*. On the other hand, we pass a user's information need to the WM-Wikifier [10], it returns a weighted list of most important topics (Wikipedia articles). For example, if our information need is NO.10.2452/702AH (mentioned before in Sec. 1), WM-Wikifier returns the following weighted list of articles:

- 0.9061: Chromotherapy[4]
- 0.1255: Color[5]
- 0.1193: Therapy[6]

[3] There are other methods that implicitly map a fragment of text into a weighted list of Wikipedia concepts such as ESA [4].

[4] http://en.wikipedia.org/wiki/Chromotherapy

[5] http://en.wikipedia.org/wiki/Color

[6] http://en.wikipedia.org/wiki/Therapy

4 Structured Query Construction

If we can map an unstructured user's information need to a weighted list of Wikipedia concepts, what can we do with these concepts?!

It can help us to move from unstructured, limited and noisy text to structured, well-known and accurate concepts. It's a break through step in IR and NLP. In our experiments, we utilize the WM-Wikifier [10] algorithm in order to extract a weighted list of Wikipedia concepts and mine translation and synonyms of these concepts from Wikipedia knowledge-base to construct an equivalent structure query. *Indri Structured Query Language* [13] is an efficient and verified query language in IR that supports an efficient retrieval model [7]. It can handle both simple keyword queries and extremely complex queries include complex phrase matching, synonyms, weighted expressions and Boolean filtering, among others. In this paper, we utilize some of them. See [13] for more details about Indri structured query language. In this section, we seek to evaluate different configuration of structured query construction. In the working note [5], the following configurations have been evaluated[7]:

– "SIMTR": Wikipedia contains articles in more than 250 natural languages. Each article link to equivalent one in other languages. After extracting concepts from unstructured user's information need, we can utilize the translation links in Wikipedia in order to translate each concept and treat them as synonyms. The query model for "SIMTR" is as follow:

```
#combine( <title> <description> #syn(#1(EN) #1(FR) #1(GE)) )
```

– "SIMEXT": For covering various equivalents, misspelling, and..., we leverage anchor titles in Wikipedia and treat all anchors as synonym. This assumption construct the following structure query [8]:

```
#combine( <title> <description>
  #syn(#1(EN) #1(FR) #1(GE) <Anchors List>))
```

The performance comparison of these methods describe in Table 2. See [5] for more details and samples about the above configurations. In this paper we present more investigations. We can embed "SIMEXT" with the weights of each concept. In this case, we have to leverage new operators such as "#weight" and "#wsyn" instead of "#combine" and "#syn". This configuration is more complicated than previous ones and take more time to run but more efficient since utilize the importance of each concept or synonym. The last configuration is "SIMHUM" which participate the user in the retrieval process. For each query, we asked the user to select the most important concepts and synonyms through a web interface. Table 2 compare the perfomance of different configurations and shows that the selecting and weighting strategies in "SIMEXT" and "SIMWEXT" were good.

[7] These notations are related to the first column in Table 2.
[8] An example of "SIMEXT" is described in Sec. 1 for a sample query.

Table 2. Comparsion between the Indri baseline ("SIM") and different configuration of our approach ("SIMTR", "SIMEXT", "SIMWEXT", "SIMHUM") and also a state of the art related work ("SD", "FD") [8]

Run Name	Relevant-Retrieved	MAP	NDCG	R-PREC	Run Time (sec)
Baseline [7]	1518/2527	0.2013	0.4635	0.2350	5
SD [8]	1522/2527	0.1980	0.4634	0.2181	112
FD [8]	1542/2527	0.2061	0.4714	0.2416	657
SIMTR	1645/2527	0.2390	0.5132	0.2688	12
SIMEXT	1724/2527	0.2462	0.5306	0.2794	17
SIMWEXT	1778/2527	0.2438	0.5296	0.2739	22
SIMHUM	1763/2527	0.2459	0.5316	0.2734	–

5 Evaluation and Comparison

This section puts the performance of our methodology in the context of state of the art prior work. Metzler and Croft [8] proposed a general, formal framework for modeling term dependencies via Markov random fields. This is a pure mathematic method that is completely different with our knowledge based approach so comparing the performance may be unfair. But since the aim of both approaches is to create a more expressive structured query for an unstructured information need, we've compared them here. Two variants of the model are described, where each captures different dependencies between query terms. The sequential dependence variant ("SD") assumes certain dependencies exist between adjacent query terms and the full dependence model ("FD") makes no independence assumptions and attempts to capture dependencies that exist between every subset of query terms. The following is a generated structured query by the sequential dependence variant for the sample information need mentioned in Sec. 1:

```
#weight( 0.5 #combine(colour therapeutic therapy)
0.25 #combine(#1(therapeutic therapy)#1(colour therapeutic))
0.25 #combine(#uw8(therapeutic therapy)#uw8(colour therapeutic)))
```

Table 2 compares the effectiveness of different variants of markov random fields [8] with different configuration of our proposed approach. The knowledge based methods significantly get a higher precision and recall while they are more scalable (the retrieval time of the proposed method is significantly less than the related work).

6 Conclusion and Discussion

In this paper, we proposed an efficient approach for extracting relevant concepts and a vocabulary of synonyms, translations, various equivalents that all of them are embedded in a structured query. We leverage Wikipedia as our knowledge base and Indri as Structured Query Language and retrieval model. Our approach is similar to query modification techniques. These techniques (such as query

expansion) suffer from a problem so-called *"Query Drift"*. It means that although by modifying a query we can get more relevant documents but it maybe hurt the precision. Our experiments over TEL corpus show that this method is an efficient and robust approach that significantly improves both precision and recall. We believe that our method is a good potential to apply on the Web search engines. For example, take a look at the following query[9]:

```
Title: Modern Persian Language,
Desc: Retrieve publications providing instructions on
learning or teaching modern/contemporary Persian.
```

The structured query by "SIMEXT" is as follow:

```
#weight(0.3 #combine(modern teaching instructions persian
contemporary learning language)  0.7 #syn(farsi #1(persian languages)
#1(farsi salis language) #1(modern perisan) persian #1(modern persian)
#1(modern persian language) #1(parsi language) #1(farsi language)
#1(persian language) #1(persische sprache) ))
```

"Farsi" or "Parsi" are informal equivalents of "Modern Persian Language" that it couldn't nowise understand from the original query. Using these informal equivalent on the Web search engines is very important evidence. For another example, take a look at the following structured query:

```
#combine(colour therapy therapeutic
  #syn(chromotherapy farbtherapi colourology #1(color therapy))
  #syn(color couleur farb colour colors colours couleur)
  #syn(therapi thrap therapi treatment therapie therapy))
```

As you see, without applying a complicated stemmer in our multilingual environment (TEL corpus), our extracted vocabulary from anchor titles can cover most of them efficiently. For example, in the structured query, "color" and "colour" are synonyms. On the other hand, the extracted vocabulary in our approach contains erratum, stemmed equivalent and synonyms of each concept. All of them are embedded in the structured query.

Although our proposed approach is interesting in some respects but comparing it with top ranked systems at CLEF2009 reveals that it needs to be refined to well adapted to the CLEF Ad Hoc track. We are optimistic about the potential of our approach in the Web search. So we are interested in performing similar experiments on the Web. Also further analysis of the Wikipedia Concept Graph to elicit more relevant concepts will be an effective future work.

References

1. Agirre, E., Di Nunzio, G.M., Ferro, N., Mandl, T., Peters, C.: CLEF 2008 Ad hoc track overview. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 15–37. Springer, Heidelberg (2009)

[9] 10.2452/733AH.

2. Callan, J.P., Croft, W.B., Broglio, J.: Trec and tipster experiments with inquery. Inf. Process. Manage. 31(3), 327–343 (1995)
3. Ferro, N., Peters, C.: CLEF 2009 Ad Hoc Track Overview: TEL & Persian Tasks. In: Workshop on Cross-Language Information Retrieval and Evaluation, Corfu, Greece (2009)
4. Gabrilovich, E., Markovitch, S.: Wikipedia-based Semantic Interpretation for Natural Language Processing. J. Artificial Intelligence Research (JAIR) 34, 443–498 (2009)
5. Jadidinejad, A.H., Mahmoudi, F.: Query Wikification: Mining Structured Queries From Unstructured Information Needs using Wikipedia-based Semantic Analysis. Technical report, CLEF2009 Working Notes (2009)
6. Medelyan, O., Milne, D., Legg, C., Witten, I.H.: Mining meaning from Wikipedia. International Journal of Human-Computer Studies 67(89), 716–754 (2009)
7. Metzler, D., Croft, W.B.: Combining the language model and inference network approaches to retrieval. Inf. Process. Manage. 40(5), 735–750 (2004)
8. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: SIGIR 2005: Proceedings of the 28th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, pp. 472–479 (2005)
9. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: 16th ACM Conference on Information and Knowledge Management, pp. 233–242. ACM, New York (2007)
10. Milne, D., Witten, I.H.: Learning to link with wikipedia. In: 17th ACM Conference on Information and Knowledge Management, pp. 509–518. ACM, New York (2008)
11. Milne, D., Witten, I.H.: An Open-Source Toolkit for Mining Wikipedia. To be Announced, http://wikipedia-miner.sourceforge.net
12. Ogilvie, P., Callan, J.: Experiments Using the Lemur Toolkit. In: 10th Text Retrieval Conference (TREC-10), pp. 103–108. TREC (2002)
13. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A languagemodel based search engine for complex queries (extended version). Technical report, University of Massachusetts (2005)

Sampling Precision to Depth 10000 at CLEF 2009

Stephen Tomlinson

Open Text Corporation
Ottawa, Ontario, Canada
stomlins@opentext.com
http://www.opentext.com/

Abstract. We conducted an experiment to test the completeness of the relevance judgments for the monolingual German, French, English and Persian (Farsi) information retrieval tasks of the Ad Hoc Track of the Cross-Language Evaluation Forum (CLEF) 2009. In the ad hoc retrieval tasks, the system was given 50 natural language queries, and the goal was to find all of the relevant documents (with high precision) in a particular document set. For each language, we submitted a sample of the first 10000 retrieved items to investigate the frequency of relevant items at deeper ranks than the official judging depth of 60 for German, French and English and 80 for Persian. The results suggest that, on average, the percentage of relevant items assessed was less than 62% for German, 27% for French, 35% for English and 22% for Persian.

1 Introduction

Open Text eDOCS SearchServer™ is a toolkit for developing enterprise search and retrieval applications. The eDOCS SearchServer kernel is also embedded in various components of the Open Text eDOCS Suite[1].

The eDOCS SearchServer kernel works in Unicode internally [4] and supports most of the world's major character sets and languages. The major conferences in text retrieval experimentation (CLEF [1], NTCIR [5] and TREC [8]) have provided judged test collections for objective experimentation with the SearchServer kernel in more than a dozen languages.

This paper describes an experiment conducted with the eDOCS SearchServer kernel (experimental post-6.0 builds) for testing the completeness of the relevance judgments for the monolingual German, French, English and Persian information retrieval tasks of the Ad Hoc Track of the Cross-Language Evaluation Forum (CLEF) 2009.

[1] Open Text eDOCS SearchServer and Open Text eDOCS Suite are trademarks or registered trademarks of Open Text Corporation in the United States of America, Canada, the European Union and/or other countries. This list of trademarks is not exhaustive. Other trademarks, registered trademarks, product names, company names, brands and service names mentioned herein are property of Open Text Corporation or other respective owners.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 78–85, 2010.

2 Methodology

2.1 Data

The CLEF 2009 Ad Hoc Track document sets were the same as used in 2008. They consisted of XML-tagged records or documents in 4 different languages: German, French, English and Persian (also known as Farsi). For German, French and English, the records were library catalog cards (bibliographic records describing publications archived by The European Library (TEL)). For Persian, the documents were newspaper articles (Hamshahri corpus of 1996-2002). Table 1 gives the collection sizes.

Table 1. Sizes of CLEF 2009 Ad Hoc Track Test Collections

Code	Language	Text Size (uncompressed)	Documents	Topics	Rel/Topic
DE	German	1,306,492,248 bytes	869,353	50	31 (lo 3, hi 86)
EN	English	1,208,383,351 bytes	1,000,100	50	51 (lo 8, hi 235)
FA	Persian	628,471,252 bytes	166,774	50	89 (lo 8, hi 266)
FR	French	1,362,122,091 bytes	1,000,100	50	37 (lo 2, hi 120)

The CLEF organizers created 50 natural language "topics" (numbered 701-750 for German, French and English and 601-650 for Persian) and translated them into many languages. Sometimes topics are discarded for some languages because of a lack of relevant documents (though that did not happen this year). Table 1 gives the final number of topics for each language and their average number of relevant documents (along with the lowest and highest number of relevant documents of any topic). For more information on the CLEF test collections, please see the track overview paper [2].

2.2 Base Run

Our base run for each language retrieved the top-10,000 ranked documents for each topic.

For each topic, the query was formed from a Boolean-OR of the words in the Title and Description fields of the topic. (For German, French and English, common instruction words such as "find", "relevant" and "document" were automatically removed before forming the query based on a word list created from looking at some older topic sets; this step was skipped for Persian because we did not have time to update our lists this year.)

The word matching was based on the stems of the terms. For German, French and English, we used the lexicon-based inflectional stemming component of SearchServer, which includes decompounding for German. For Persian, we used a stemmer that was ported from Savoy's [7].

A stopword list of common words (e.g. "the", "of") was used for each language. Our stopword list for Persian was derived from Savoy's [7].

The ranking approach was as described in [9]. Briefly, it dampened the term frequency and adjusted for document length in a manner similar to Okapi [6] and dampened the inverse document frequency using an approximation of the logarithm.

2.3 Sample Run

For each language, we created a sample run whose first 100 rows contained the following rows of the base run for the language in the following order:

```
1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
20, 30, 40, 50, 60, 70, 80, 90, 100,
200, 300, 400, 500, 600, 700, 800, 900, 1000,
2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000,
15, 25, ..., 95,
150, 250, ..., 950,
1500, 2500, ..., 9500,
125, 175, ..., 975,
1250, 1750, ..., 9750.
```

The remainder of the sample run was padded with the top-ranked remaining rows from the base run until 1000 rows had been retrieved (i.e. rows 11, 12, 13, 14, 16, ..., 962 of the base run).

This ordering (e.g. the placement of the sample from depth 10000 before the sample from depth 15) was chosen because of uncertainty of how deep the judging would be. As long as the first 37 rows were judged for each topic, we would have sampling to depth 10000 (this is because, in the above list, you can count that, after 37 samples, depth 10000 is reached). The extra sample points, if judged, would just improve the accuracy (because they are just additional sample points from the top 10000, not deeper sample points).

Our sample run for each language was submitted to the CLEF organizers for assessing in June 2009.

3 Results

When we received the relevance judgments and analyzed them in August 2009, we checked the judging depth of our sample runs. We found that the first 60 rows were judged for each topic for each German, French and English, and the first 80 rows were judged for each topic for Persian.

Tables 2, 3, 4 and 5 show the results of the sampling for each language. The columns are as follows:

– "Depth Range": The range of depths being sampled. The 11 depth ranges cover from 1 to 10000.

- "Samples": The depths of the sample points from the depth range. The samples are always uniformly spaced. They always end at the last point of the depth range. The total number of sample points (over the 11 rows of the table) adds to 60 for German, French and English and adds to 80 for Persian.
- "# Rel": The number of each type of item retrieved from the sample points over the 50 topics. The item type codes are R (relevant), N (non-relevant) and U (unjudged, of which there are always 0). An X is used when a sample point was not submitted because fewer than 10000 rows were retrieved for the topic (which just happened for a few topics). The sum of the item type counts is always 50 times the number of sample points for the depth range (because there are 50 topics for each language).
- "Precision": Estimated precision of the depth range (R/(R+N+U+X)).
- "Wgt": The weight of each sample point. The weight is equal to the difference in ranks between sample points, i.e. each sample point can be thought of as representing this number of rows, which is itself plus the preceding unsampled rows.
- "EstRel/Topic": Estimated number of relevant items retrieved per topic for this depth range. This is the Precision multiplied by the size of the depth range. Or equivalently, it is (R * Wgt) / 50.

Table 2. Marginal Precision of German Base-TD Run at Various Depths

Depth Range	Samples	# Rel	Precision	Wgt	EstRel/Topic
1-5	1, 2, ..., 5	106R, 144N, 0U	0.424	1	2.1
6-10	6, 7, ..., 10	76R, 174N, 0U	0.304	1	1.5
11-50	15, 20, ..., 50	72R, 328N, 0U	0.180	5	7.2
51-100	55, 60, ..., 100	44R, 456N, 0U	0.088	5	4.4
101-200	150, 200	5R, 95N, 0U	0.050	50	5.0
201-500	250, 300, ..., 500	5R, 295N, 0U	0.017	50	5.0
501-900	550, 600, ..., 900	5R, 395N, 0U	0.013	50	5.0
901-1000	950, 1000	0R, 100N, 0U	0.000	50	0.0
1001-3000	1500, 2000, ..., 3000	1R, 199N, 0U	0.005	500	10.0
3001-6000	3500, 4000, ..., 6000	1R, 295N, 4X	0.003	500	10.0
6001-10000	7000, 8000, ..., 10000	0R, 189N, 11X	0.000	1000	0.0

Because each sample point is at the deep end of the range of rows it represents, the sampling should tend to underestimate precision for each depth range (assuming that precision tends to fall with depth, which appears to be the case for all 4 languages).

Table 6 shows the sums of the estimated number of relevant items per topic over all depth ranges in its first row (i.e. it is the sum of the EstRel/Topic entries in the last column of the corresponding table from Tables 2-5). The official number of relevant items per topic for each language is listed in the second row. The final row of the table just divides the official number of relevant items by the

Table 3. Marginal Precision of French Base-TD Run at Various Depths

Depth Range	Samples	# Rel	Precision	Wgt	EstRel/Topic
1-5	1, 2, ..., 5	90R, 160N, 0U	0.360	1	1.8
6-10	6, 7, ..., 10	88R, 162N, 0U	0.352	1	1.8
11-50	15, 20, ..., 50	89R, 311N, 0U	0.223	5	8.9
51-100	55, 60, ..., 100	60R, 440N, 0U	0.120	5	6.0
101-200	150, 200	8R, 92N, 0U	0.080	50	8.0
201-500	250, 300, ..., 500	15R, 285N, 0U	0.050	50	15.0
501-900	550, 600, ..., 900	5R, 395N, 0U	0.013	50	5.0
901-1000	950, 1000	2R, 98N, 0U	0.020	50	2.0
1001-3000	1500, 2000, ..., 3000	3R, 196N, 1X	0.015	500	30.0
3001-6000	3500, 4000, ..., 6000	4R, 288N, 8X	0.013	500	40.0
6001-10000	7000, 8000, ..., 10000	1R, 189N, 10X	0.005	1000	20.0

Table 4. Marginal Precision of English Base-TD Run at Various Depths

Depth Range	Samples	# Rel	Precision	Wgt	EstRel/Topic
1-5	1, 2, ..., 5	128R, 122N, 0U	0.512	1	2.6
6-10	6, 7, ..., 10	112R, 138N, 0U	0.448	1	2.2
11-50	15, 20, ..., 50	107R, 293N, 0U	0.268	5	10.7
51-100	55, 60, ..., 100	44R, 456N, 0U	0.088	5	4.4
101-200	150, 200	7R, 93N, 0U	0.070	50	7.0
201-500	250, 300, ..., 500	12R, 288N, 0U	0.040	50	12.0
501-900	550, 600, ..., 900	15R, 385N, 0U	0.037	50	15.0
901-1000	950, 1000	2R, 98N, 0U	0.020	50	2.0
1001-3000	1500, 2000, ..., 3000	4R, 196N, 0U	0.020	500	40.0
3001-6000	3500, 4000, ..., 6000	1R, 297N, 2X	0.003	500	10.0
6001-10000	7000, 8000, ..., 10000	2R, 194N, 4X	0.010	1000	40.0

Table 5. Marginal Precision of Persian Base-TD Run at Various Depths

Depth Range	Samples	# Rel	Precision	Wgt	EstRel/Topic
1-5	1, 2, ..., 5	158R, 92N, 0U	0.632	1	3.2
6-10	6, 7, ..., 10	131R, 119N, 0U	0.524	1	2.6
11-50	15, 20, ..., 50	159R, 241N, 0U	0.398	5	15.9
51-100	55, 60, ..., 100	151R, 349N, 0U	0.302	5	15.1
101-200	125, 150, ..., 200	38R, 162N, 0U	0.190	25	19.0
201-500	225, 250, ..., 500	93R, 507N, 0U	0.155	25	46.5
501-900	525, 550, ..., 900	81R, 719N, 0U	0.101	25	40.5
901-1000	950, 1000	7R, 93N, 0U	0.070	50	7.0
1001-3000	1500, 2000, ..., 3000	7R, 193N, 0U	0.035	500	70.0
3001-6000	3500, 4000, ..., 6000	10R, 290N, 0U	0.033	500	100.0
6001-10000	6500, 7000, ..., 10000	9R, 388N, 3X	0.022	500	90.0

Table 6. Estimated Percentage of Relevant Items that are Judged

	DE	FR	EN	FA
Estimated Rel@10000	50.2	138.5	145.9	409.8
Official Rel/Topic	31.2	37.1	50.5	89.3
Percentage Judged	62%	27%	35%	22%

estimated number in the first 10000 retrieved (e.g. for German, 31.2/50.2=62%). This number should tend to be an overestimate of the percentage of all relevant items that are judged (on average per topic) because there may be relevant items that were not matched by the query in the first 10000 rows.

3.1 Remarks

These estimates of judging coverage for the CLEF 2009 collections are similar to last year's estimates [12] for two of the four languages (62% for German this year, 55% last year; 22% for Persian this year, 25% last year). For the other two languages, this year's estimates are substantially lower than last year's (27% for French this year, 52% last year; 35% for English this year, 53% last year). We've used similar methodology (though sometimes using different sampling depths) for other past collections, such as the CLEF 2007 Ad Hoc collections (55% for Czech, 69% for Bulgarian, 83% for Hungarian) [11], the NTCIR-7 ACLIA IR4QA collections (65% for Simplified Chinese, 32% for Traditional Chinese, 41% for Japanese) [13], the NTCIR-6 CLIR collections (58% for Chinese, 78% for Japanese, 100% for Korean) [14], and the TREC 2006 Legal and Terabyte collections (18% for TREC Legal and 36% for TREC Terabyte) [10].

The incompleteness results for German are similar to what [15] found for depth-100 pooling on the old TREC collections of approximately 500,000 documents. [15] reported that "it is likely that at best 50%-70% of the relevant documents have been found; most of these unjudged relevant documents are for the 10 or so queries that already have the most known answers." Fortunately, [15] also found for such test collections that "overall they do indeed lead to reliable results." [3] also considers the "levels of completeness" in some older TREC collections to be "quite acceptable" even though additional judging found additional relevant documents.

For English, French and Persian, the judging coverage appears to have been relatively shallow (35%, 27% and 22% respectively based on the sampling experiment). It may be advisable to conduct a "system omission" study on these collections (like the one described in [15]) which may indicate whether or not the collections are likely to give reliable results for systems that did not contribute to the pooling.

3.2 Error Analysis

We should note that our sampling was very coarse at the deeper ranks, e.g. for French, 1 relevant item out of 200 samples in the 6001-10000 range led to an estimate of 20 relevant items per topic in this range. If the sampling had turned up 0 or 2 relevant items, a minor difference, the estimate would have been 0 or 40 relevant items per topic in this range, leading to a substantially different sum (118.5 or 158.5 instead of 138.5). We leave the computation of confidence intervals for our estimates, along with analysis of the variance across topics, as future work.

4 Conclusions

We conducted an experiment to test the completeness of the relevance judgments for the monolingual German, French, English and Persian information retrieval tasks of the Ad Hoc Track of the Cross-Language Evaluation Forum (CLEF) 2009. For each language, we submitted a sample of the first 10000 retrieved items to investigate the frequency of relevant items at deeper ranks than the official judging depth of 60 for German, French and English and 80 for Persian. Based on the results, we estimated that the percentage of relevant items assessed was less than 62% for German, 27% for French, 35% for English and 22% for Persian. For German, these levels of completeness are in line with the estimates that have been made for some past test collections which are still considered useful and fair for comparing retrieval methods. For English, French and Persian, the completeless levels are lower than usual. For any test collection, it is prudent to conduct a "system omission" study (like the one described in [15]) which may indicate whether or not the collection is likely to give reliable results for systems that did not contribute to the pooling. Such a study would be particularly advisable for the English, French and Persian collections.

References

1. Cross-Language Evaluation Forum web site, http://www.clef-campaign.org/
2. Ferro, N., Peters, C.: CLEF 2009 Ad Hoc Track Overview: TEL & Persian Tasks. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 13–35. Springer, Heidelberg (2010)
3. Harman, D.K.: The TREC Test Collections. In: TREC: Experiment and Evaluation in Information Retrieval (2005)
4. Hodgson, A.: Converting the Fulcrum Search Engine to Unicode. In: Sixteenth International Unicode Conference (2000)
5. NTCIR (NII-NACSIS Test Collection for IR Systems) Home Page, http://research.nii.ac.jp/~ntcadm/index-en.html
6. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at TREC-3. In: Proceedings of TREC-3 (1995)
7. Savoy, J.: CLEF and Multilingual information retrieval resource page, http://www.unine.ch/info/clef/

8. Text REtrieval Conference (TREC) Home Page, http://trec.nist.gov/
9. Tomlinson, S.: Bulgarian and Hungarian Experiments with Hummingbird SearchServer™ at CLEF 2005. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, pp. 194–203. Springer, Heidelberg (2006)
10. Tomlinson, S.: Experiments with the Negotiated Boolean Queries of the TREC 2006 Legal Discovery Track. In: Proceedings of TREC 2006 (2006)
11. Tomlinson, S.: Sampling Precision to Depth 10000 at CLEF 2007. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 57–63. Springer, Heidelberg (2008)
12. Tomlinson, S.: Sampling Precision to Depth 10000 at CLEF 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 163–169. Springer, Heidelberg (2009)
13. Experiments in Finding Chinese and Japanese Answer Documents at NTCIR-7. In: Proceedings of NTCIR-7 (2008)
14. Tomlinson, S.: Sampling Precision to Depth 9000: Evaluation Experiments at NTCIR-6. In: Proceedings of NTCIR-6 (2007)
15. Zobel, J.: How Reliable are the Results of Large-Scale Information Retrieval Experiments? In: SIGIR 1998, pp. 307–314 (1998)

Multilingual Query Expansion for CLEF Adhoc-TEL

Ray R. Larson

School of Information
University of California, Berkeley, USA
ray@sims.berkeley.edu

Abstract. In this paper we will briefly describe the approaches taken by the Cheshire (Berkeley) Group for the CLEF Adhoc-TEL 2009 tasks (Mono and Bilingual retrieval). Recognizing that many potentially relevant documents in each of the TEL sub-collections are in other languages, we tried to use multiple translations of the topics for searching each subcollection, combined into a single query. Overall this strategy performed very poorly compared to the the basic monolingual approach used last year (and repeated for one run in each language this year). Once again this year we used probabilistic text retrieval based on logistic regression and incorporating blind relevance feedback for all of the runs. All translation for bilingual tasks was performed using the LEC Power Translator PC-based MT system. Our results this year, however, were surprising poor compared to last year's results. Additional analysis has shown that, for some cases, unexpected hyphenations in the machine translation and untranslated words were to blame.

1 Introduction

Each the collections used in the CLEF Adhoc TEL track are considered to be "mainly" in a particular language (English for BL, French for BNF, and German for ONB), according to the language codes of the records, only about half of each collection was in that main language, with virtually all other languages represented by one or more entries in one or another of the collections. German, French, English, and Spanish records were available in all of collections. This overlap of languages presents an interesting multilingual search (and evaluation) problem, and we attempted to address it this year by using tranlations of topics into each of the other languages and combining those translations with the original topic in some of our submissions.

This short paper concentrates on the retrieval algorithms and evaluation results for Berkeley's official submissions for the Adhoc-TEL 2009 track, and our analysis of problems in the submitted runs. All of the submitted runs were automatic without manual intervention in the queries (or translations). We submitted nine Monolingual runs (three German, three English, and three French) and 12 Bilingual runs (four for each target language German, English and French, with both expanded and unexpanded topics).

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 86–89, 2010.

This short paper first describes the processing used for the submitted runs. We then examine the results obtained for our official runs and examine the sources of errors in those runs. Finally, we present some conclusions and future directions for Adhoc-TEL participation.

2 Retrieval Algorithms and Indexing Approaches

Since this is not our main paper, we have forgone detailed discussion of the algorithms used for this track, which are essentially the same Logistic Regression algorithm with Blind Feedback as used in other CLEF Evaluations in previous years and described in [2,1], or in the notebook paper for this year.

In the remainder of this section we describe the specific approaches taken for our submitted runs for the Adhoc-TEL task. First we describe the indexing and term extraction methods used, and then the search features we used for the submitted runs.

2.1 Indexing and Term Extraction

The Cheshire II system uses the XML structure of the documents to extract selected portions for indexing and retrieval. Any combination of tags can be used to define the index contents.

Table 1. Cheshire II Indexes for Adhoc-TEL 2006

Name	Description	Content Tags	Used
recid	Document ID	id	no
names	Author Names	dc:creator, dc:contributor	no
title	Item Title	dc:title, dcterms:alternate	no
topic	Content Words	dc:title, dcterms:alternate dc:subject, dc:description	yes
anywhere	Entire record	record	no
date	Date of Pub.	dcterms:issued	no
lang	Language	dc:language	no
subject	Subject terms	dc:subject	no

Table 1 lists the indexes created by the Cheshire II system for the Adhoc-TEL database and the document elements from which the contents of those indexes were extracted. The "Used" column in Table 1 indicates whether or not a particular index was used in the submitted Adhoc-TEL runs. As the table shows we used only the topic index, which contains most of the content-bearing parts of records, for all of our submitted runs. These tables and the indexes extracted are identical to last year's for Adhoc TEL.

For all indexing we used language-specific stoplists to exclude function words and very common words from the indexing and searching. The German language

Table 2. Submitted Adhoc-TEL Runs

Run Name	Description	Type	MAP
MODET2FB	Monolingual German	TD auto	0.1478
MOENT2FB	Monolingual English	TD auto	0.3267
MOFRT2FB	Monolingual French	TD auto	0.2070
BIENDET2FB	Bilingual English⇒German	TD auto	0.1031
BIFRDET2FB	Bilingual French⇒German	TD auto	0.0991
BIDEENT2FB	Bilingual German⇒English	TD auto	0.2238
BIFRENT2FB	Bilingual French⇒English	TD auto	0.2478
BIDEFRT2FB	Bilingual German⇒French	TD auto	0.1652
BIENFRT2FB	Bilingual English⇒French	TD auto	0.1677

runs *did not* use decompounding in the indexing and querying processes to generate simple word forms from compounds. The Snowball stemmer was used by Cheshire for language-specific stemming.

2.2 Search Processing and Results

Searching the Adhoc-TEL collection using the Cheshire II system involved using TCL scripts to parse the topics and submit the title and description from the topics. For monolingual search tasks we used the topics in the appropriate language (English, German, and French), for bilingual tasks the topics were translated from the source language to the target language using the LEC Power Translator PC-based machine translation system.

For query expansion in the monolingual tasks we took two approaches. The first used the topic in the specific language as a basis for machine translation to the other main languages (e.g. for English, the English topics were translated to French and German) and the translations were added to the topic. The second used the supplied monolingual topics in the other main languages (e.g., for English, the monolingual French and German topics were added to the English).

Query expansion in the bilingual tasks added the source topics from the translation and an additional translation of the topics to the other main language (e.g., for English topics translated to German, the original English was added to the translated German and an English to French translation was also added). In effect, the expanded monolingual and bilingual topics were actually multilingual topic descriptions.

The scripts for each run submitted the topic elements as they appeared in the topic or expanded topic to the system for TREC2 logistic regression searching with blind feedback. Both the "title" and "description" topic elements were combined into a single probabilistic query and searched using the "topic" index as described in Table 1.

The summary results (as Mean Average Precision) for the submitted bilingual and monolingual runs for English, German and French are shown in Table 2 for only the unexpanded topics, all submitted runs and the Recall-Precision curves for these runs are not shown, but may be found in the notebook paper.

Both of the query expansion methods attempted proved to provide worse results than the unexpanded queries.

Once again we obtained particularly poor performance in monolingual German, due in part to our lack of support for decompounding (affecting many topics this year).

3 Conclusions and Discussion

Our overall results this year compared poorly with others, which was a bit of a surprise considering the how the same approach fared last year. We conducted some analyses to try to determine the causes of variation between last year and this and the causes of failure for various topics. One very obvious change was that a new version of the MT software was used this year. We found that that translations from German often had compound terms included in the translation as hyphenated terms (e.g., "color-therapy" for "Farbentherapie"). To see what effect this might have had in some runs we translated the hyphens in such cases to spaces and reran some experiments. The results of this re-test showed that for the German to English bilingual task we were able to obtain a MAP of 0.2613 compared to 0.2238 in our official results with this simple change. This, however does not explain the relative failures in monolingual results. This turned out to be an encoding mismatch in the German and French databases with the version of the Snowball stemmer that we used. Effectively we had all of the data encoded as UTF-8, but the stemmer parsing for ISO-8859-1. This meant that stemming process was ineffective and identically inflected stems only were matched in retrieval. This still doesn't explain the reduction in MAP for our Monolingual English runs when compared to last year. We can only conclude that methods used by other groups this year are more effective in these databases.

References

1. Larson, R.R.: Cheshire at geoclef 2007: Retesting text retrieval baselines. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 811–814. Springer, Heidelberg (2008)
2. Larson, R.R.: Experiments in classification clustering and thesaurus expansion for domain specific cross-language retrieval. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 188–195. Springer, Heidelberg (2008)

Experiments with N-Gram Prefixes on a Multinomial Language Model versus Lucene's Off-the-Shelf Ranking Scheme and Rocchio Query Expansion (TEL@CLEF Monolingual Task)

Jorge Machado, Bruno Martins, and José Borbinha

Departmento de Engenharia Informática, Technical University of Lisbon, Portugal.
{jorge.r.machado,bruno.martins,jose.borbinha}@ist.utl.pt

Abstract. We describe our participation in the TEL@CLEF task of the CLEF 2009 ad-hoc track, where we measured the retrieval performance of LGTE, an index engine for Geo-Temporal collections which is mostly based on Lucene, together with extensions for query expansion and multinomial language modelling. We experiment an N-Gram stemming model to improve our last year experiments which consisted in combinations of query expansion, Lucene's off-the-shelf ranking scheme and the ranking scheme based on multinomial language modeling. The N-Gram stemming model was based in a linear combination of N-Grams, with N between 2 and 5, using weight factors obtained by learning from last year topics and assessments. The Rocchio ranking function was also adapted to implement this N-Gram model. Results show that this stemming technique together with query expansion and multinomial language modeling both result in increased performance.

1 Introduction

One task of the ad-hoc track at the 2009 edition of the Cross Language Evaluation Forum (CLEF) addresses the problem of searching and retrieving relevant items from collections of bibliographic records from The European Library (TEL@CLEF). Three target collections were provided, each corresponding to a monolingual retrieval task where we participated: the TEL Catalogue records in English (Copyright British Library), the TEL Catalogue records in French (Copyright Bibliothèque Nationale de France) and finally the TEL Catalogue records in German (Copyright Austrian National Library). The evaluation task aimed at investigating the best approaches for retrieval from library catalogues, where the information is frequently very sparse and often stored in unexpected languages. This paper describes the participation of the Technical University of Lisbon in the TEL@CLEF task. Our experiments aimed at measuring the retrieval performance of the LGTE[1] tool [8] [4], the IR service of DIGMAP[2], using stemming techniques to render the system language independent

[1] http://code.google.com/p/digmap/wiki/LuceneGeoTemporal
[2] http://www.dgmap.eu

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 90–97, 2010.

and robust with degraded texts resulting from OCR processes. If successful, the ultimate goal of the project is to become fully integrated into The European Library which aims to index OCR texts in multiple languages.

Like last year in CLEF, we experimented with combinations of query expansion, Lucene's off-the-shelf ranking scheme and the ranking scheme based on multinomial language modeling. However, this year we also included an N-Gram model consisting in a linear combination of independent indexes, each containing stemmed tokens of different grams. The technique was proposed by Parapar in [9] and aims to improve retrieval in degraded collections. Our main objective was to have a language independent model which also could be used with bibliographic metadata records, which of course are not degraded. This paper is structured as follow: first we review the related word in ranking schemes used in this experiment and the related work with the Rocchio's algorithm. Second we describe our ranking scheme and the modifications purposed for the Rocchio's algorithm in order to take benefit of our scheme. In third place we describe our experimental story and discuss the obtained results. Finally we present conclusions.

2 Related Work

The underlying IR system used in our submissions is based on Lucene[3], together with a multinomial language modeling extension developed at the University of Amsterdam, a linear combination of scores calculated in independent indexes of words stemmed with an N-Gram technique, and finally a query expansion extension developed by Neil Rubens. The following subsections detail these components.

2.1 Lucene's Off-the-Shelf Ranking Scheme

We started with Lucene's off-the-shelf retrieval model. For a collection D, document d and query q, the ranking score is given by terms of term frequency (*tf*), inverse document frequency (*idf*) and normalization factors (*norm* and *coord*):

$$ranking\ (q,d) = \sum_{t \in q} \frac{tf_{t,q} \cdot idf_t}{norm_q} \cdot \frac{tf_{t,d} \cdot idf_t}{norm_d} \cdot coord_{q,d} \cdot weight_t \tag{1}$$

The fields in formula 1 are detailed below by formula 2:

$$tf_{t,X} = \sqrt{termFrequency(t,X)}, \qquad idf_t = 1 + \log \frac{|D|}{documentFrequency(t,D)}, \qquad norm_q = \sqrt{\sum_{t \in q} tf_{t,q} \cdot idf_t^2}, \qquad norm_d = \sqrt{|d|}, \qquad coord_{q,d} = \frac{|q \cap d|}{|q|} \tag{2}$$

Lucene has been extensively used in previous editions of the CLEF, NTCIR and TREC joint evaluation experiments.

[3] http://lucene.apache.org

2.2 Lucene Extension Based on Multinomial Language Modeling

We experimented with an extension to Lucene that implements a retrieval scheme based on estimating a language model (LM) for each document, using the formula described by Hiemstra [2]. This extension was developed at the Informatics Institute of the University of Amsterdam[4]. For any given query, it ranks the documents with respect to the likelihood that the document's LM generated the query:

$$ranking(d,q) = P(d \mid q) \propto P(d) \cdot \prod_{t \in q} P(t \mid d) \tag{3}$$

In the formula, d is a document and t is a term in query q. The probabilities are reduced to rank-equivalent logs of probabilities. To account for data sparseness, the likelihood $P(t|d)$ is interpolated using Jelinek-Mercer smoothing:

$$P(d \mid q) = P(d) \cdot \prod_{t \in q} ((1-\lambda) \cdot P(t \mid D) + \lambda \cdot P(t \mid d)) \tag{4}$$

In the formula, D is the collection and λ is a smoothing parameter (in our experiments it was set to the default value of 0.15). The model needs to estimate three probabilities: the prior probability of the document, $P(d)$; the probability $P(t|d)$ of observing a term in a document, and the probability $P(t|D)$ of observing the term in the collection. Assuming the query terms to be independent, and using a linear interpolation of a document model and a collection model to estimate the probability of a query term, the probabilities can be estimated using maximum likelihood estimates:

$$P(t \mid d) = \frac{termFrequency(t,d)}{|d|} \qquad P(d) = \frac{|d|}{\sum_{d' \in D} |d'|}$$

$$P(t \mid D) = \frac{documentFrequency(t,D)}{\sum_{t' \in D} documentFrequency(t',D)} \tag{5}$$

This language modeling approach has been used in past experiments within the CLEF, NTCIR and TREC joint evaluation campaigns – see for example [6].

2.3 Rocchio Query Expansion

The fact that there are frequently occurring spelling variations and synonyms for any query term degrades the performance of standard techniques for ad-hoc retrieval. To overcome this problem, we experimented with the method for pseudo feedback query expansion proposed by Rocchio [3]. The Lucene extension from the LucQE project[5] implements this approach. On test data from the 2004 TREC Robust Retrieval Track, LucQE achieved a MAP score of 0.2433 using Rocchio query expansion. Assuming that the top D documents returned for an original query q_i are relevant, a better query q_i+1 can be given by the terms resulting from the formula below:

$$q_{i+1} = \alpha \cdot q_i + \frac{\beta}{|D|} \cdot \sum_{d_r \in D} termWeight(d_r) \tag{6}$$

[4] http://ilps.science.uva.nl/Resources/
[5] http://lucene-qe.sourceforge.net/

In the formula, α and β are tuning parameters. In our experiments, they were set to the default values of, respectively, 1.0 and 0.75. The system was trained through experiments with the 2008 AdHoc topics and relevance judgments. We found an optimal value of 64 expansion terms for English topics and 40 expansion terms for French and German topics. The terms were extracted from the highest ranked documents (i.e. the $|D|$ parameter) from the original query q_i. With the training we obtain optimal values using 7 documents for English and French and 8 documents for the German collection.

2.4 Linear Combination of N-Grams

The stemming technique based on N-Grams is very popular with texts produced from OCR (Optical Character Recognition) processes, because they often contain errors. This technique consists in tokenizing the words with a sliding window into tokens of size N, with N assuming several sizes. This process is applied both in documents and queries to increase retrieval performance. Recent experiments related in [9] by Parapar demonstrate that using independent N-Grams indexes, for example from 2 to 5 grams, and combining the individual ranks in a linear combination, can improve the results when we find good parameter values to weight each independent score. The final score is illustrated by the formula 7, as introduced in [9].

$$s(d) = \alpha \cdot sterm(d) + \beta \cdot s5gram(d) + \gamma \cdot s4gram(d) + \delta \cdot s3gram(d) + \varepsilon \cdot s2gram(d) \quad (7)$$

In this formula d is the document, *s[N]term* is the score of that term in the index of grams with size N. Parameters α, β, γ, δ and ε are the weights assigned to each independent score.

3 Ranking Scheme

The following subsections detail how we adapted the ranking scheme based on the combination of N-Grams for bibliographic records and also the modifications in Rocchio query expansion algorithm in order to take benefit of our ranking scheme.

3.1 N-Gram Ranking Scheme

The original N-Grams stemming, which tokenizes the words with a sliding window, does not fit our problem very well because our records were not obtained from OCR processes (so we don't have character errors). On other hand using this technique makes the stemming phase a language independent process, which was our main focus. For that reason, we used a simplistic approach for the N-Grams model which consists of suffix removal starting from character N+1. We used an "N-length stemming" where N is the size of the indexed prefix (e.g. *stem-5("retrieval") = "retri"*). We tokenized our terms in five different ways, each producing a different index file. We created four indexes, for the cases of 2-grams, 3-grams, 4-grams and 5-grams, and another with the original terms. As an example, let us consider a document with the word "retrieval". That document will be indexed as follows: originalTerms: *retrieval*, 5-grams: *retri*, 4-grams: *retr*, 3-grams: *ret*, 2-grams: *re*. Referring to the weight

parameters presented in previous section our system was trained through experiments with CLEF 2008 AdHoc topics and relevance judgments to optimize the Mean Average Precision (MAP). Table 1 shows the optimal values for formula 7 factors in each collection. We found that bi-grams worsen the results so we set their weight to zero in the three evaluated collections.

Table 1. Weight values found for each index using MAP in 2008 relevance judgments

Lanuage	α	β	γ	δ
English	0.45	0,27	0,25	0,03
French	0.53	0,24	0,22	0,01
German	0,55	0,23	0,21	0,01

3.2 N-Grams and Rocchio Query Expansion

In order to deal with N-Gram prefix stemming we had to adapt the Rocchio formula. Originally, the Rocchio algorithm calculates the ranking for the terms of the top documents with the formula (7) and selected, for the expanded query, the highest ranked terms boosting them in the final query with the obtained rank. Our problem was how to do that considering that we have five indexes instead of one. Three techniques were experimented but only the third one improved the results. The first and second attempt can be found in the CLEF 2009 Working Notes version of this paper on the CLEF website. Our best approach consisted in the following steps. For each one of our top ranked documents *D* we proceeded as follows: **First** of all, using each one of the 5 independent indexes ({2,3,4,5}grams plus original terms index), the system scored all the document terms *t* present in those indexes using the follow formula:

$$score(term, d, D, pos) = TF(term, d) \cdot IDF(term, D) \cdot decay(pos) \cdot weight(term \rightarrow index) \quad (8)$$

In this formula *D* is the collection, *decay(pos)* is the decay factor related with document *d* position in the retrieved list. The *weight* is the factor found for that term index (Table 1). **Second**, the scored terms from all the 5 indexes were sorted in one unique list, independently of the source index. **Finally,** we created the expanded query using the original terms of the query, boosted by 1, plus the top ranked terms in the sorted list boosted with the score of the term. This method weakened the tokens from less weighted indexes like 2-Grams and 3-Grams. The result was that tokens from weaker indexes could only be picked if they were very relevant in their own indexes. Expanded queries were mainly composed by tokens of 4 or 5-grams and original terms, but all queries had tokens from all indexes, even the weakest ones.

4 The Experimental Story and Obtained Results

We aimed to experiment the performance of Porter stemming technique versus the linear combination of N-Grams, with and without query expansion, using two different ranking schemes for text: the Vector Space Model and the Multinomial Language Model. Our objective was to optimize several parameters to maximize the MAP

measure using CLEF 2008 AdHoc topics and relevance judgments. For each collection (EN, FR, DE) we optimized the parameters of Rocchio technique and the weights assigned to each independent index of 3, 4, 5 grams tokens and the un-stemmed words index (original terms). The optimized values were already presented in the sections Related Work and Ranking Scheme. The optimized values were used to run the 2009 topics.

Before the indexing, the documents (i.e. the bibliographic records) and the topics were passed through the following pre-processing operations. **Field weighting** of the bibliographic records was applied using the scheme proposed by Robertson et. al [5] to weight the different document fields according to their importance. The combination used in our experiments was based on repeating the *title* field three times, the *subject* field twice and keeping the other document fields unchanged. We also **normalized** the topics and collections reducing all characters to the lowercase unaccented equivalents (i.e. "Ö" reduced to "o" and "É" to "e" etc.). We also removed **stopwords** using lists from the Snowball package[6]. We **stemmed the words** of the documents using, in first experiment, the Porter [1] stemming algorithm from the Snowball[6] package, specific to the language, and in the second experiment using tokens of length 3, 4 and 5 plus the original words in five independent indexes. The **topic processing** was fully automatic including twice the title, once the description and we didn't use the narrative. In the topics, the resulting words were also stemmed using the Porter technique or stemmed to tokens of length 3, 4 and 5 plus the original words. In the second case the queries were split into five parts, each boosted by the optimized values enumerated in Table 1 (section 2.1). Take as an example the topic "Title: *Adhoc;* Description: *information retrieval*" for English collection, the resulting query is given by:

"words:(adhoc adhoc information retrieval)^0.45 g5:(adhoc adhoc infor retri)^0.27 g4:(adho adho info retr)^0.25 g3:(adh adh inf ret)^0.03."

In the query the labels *words, g5, g4 g3* are indexes and ^*x* is a boost factor. The Lucene-LM[4] machine was adapted to calculate independently each part of the query in order to implement the linear combination of N-Grams detailed in previous sections.

5 Results

We now present the complete set of experiments for the three languages using the two text models, vector space and language model, and combining Rocchio query expansion with the two stemming approaches. Table 2 shows the obtained results in terms of the mean average precision (MAP), precision at first five results (P@5) and precision at first 10 results (P@10). In the table, VS means Vector Space Model, LM means Language Model and QE means Query Expansion.

The weighted model of N-Grams allied with the Rocchio query expansion outperforms almost all the other configurations in all languages. Using Rocchio the model performed better than the text models allied with Porter stemming technique, otherwise the performance was similar. Statisticaly comparing the MAP of baselines (runs 1 and 2) with runs 5 and 6 using grams gave less than 0,0005 for all collections except

[6] `http://snowball.tartarus.org/`

Table 2. MAP vs. MAP 2008 optimization, P@5 and P@10 for all the combinations

				English				French				German			
	Model	Stemming	QE	MAP	MAP 2008	P@5	P@10	MAP	MAP 2008	P@5	P@10	MAP	MAP 2008	P@5	P@10
1	VS	no	no	0.3403	>0.3242	0.6360	0.5200	0.2030	<0.2352	0.4400	0.3380	0.1357	>0.1290	0.3080	0.2340
2	LM	no	no	0.3496	>0.3228	0.6480	0.5260	0.2255	<0.2412	0.4680	0.4020	0.1480	<0.1523	0.3160	0.2680
3	VS	Porter	no	0.3710	<0.3789	0.6320	0.5500	0.2338	<0.2561	0.4360	0.3640	0.2372	>0.2132	0.4920	0.3720
4	LM	Porter	no	0.3829	<0.3914	0.6800	0.5480	0.2647	<0.2781	0.4760	0.3860	0.2473	>0.2326	0.5040	0.3880
5	VS	Grams	no	0.3966	>0.3750	0.6760	0.5620	0.2508	<0.2967	0.4800	0.4000	0.2439	>0.2306	0.4800	0.3680
6	LM	Grams	no	0.3902	>0.3775	0.6800	0.5500	0.2526	<0.2821	0.4960	0.4080	0.2524	>0.2266	0.4880	0.3880
7	VS	no	Rocchio	0.3712	>0.3526	0.6240	0.5400	0.2015	<0.2640	0.4320	0.3420	0.1725	<0.1875	0.3320	0.2740
8	LM	no	Rocchio	0.3778	>0.3695	0.6200	0.5420	0.2213	<0.2759	0.4280	0.3500	0.1921	>0.1913	0.3320	0.3060
9	VS	Porter	Rocchio	0.4012	>0.3980	0.6640	0.5560	0.2186	<0.2517	0.4240	0.3380	0.2810	**>0.2629**	0.5400	0.4100
10	LM	Porter	Rocchio	0.4143	**<0.4306**	0.6960	0.5920	0.2391	<0.2722	0.4240	0.3500	0.2891	>0.2586	0.5160	0.4400
11	VS	Grams	Rocchio Grams	**0.4393**	>0.4088	0.6760	0.5720	**0.2641**	**<0.3261**	0.4760	0.3880	**0.3005**	**>0.2813**	0.5080	0.4240
12	LM	Grams	Rocchio Grams	**0.4240**	**>0.4140**	0.6720	0.5680	**0.2653**	**<0.3021**	0.5120	0.4100	**0.3049**	**>0.2600**	0.5240	0.4160

for French where the result was 0.019 which was also good. Using runs 7 and 8 (Porter plus Rocchio) as baselines we tested the significance of using N-Grams plus Rocchio (runs 11 and 12) and the t-test shows significance in all results returning less than 0,015 except for Language Model in English (run 12) returning 0.2089.

These results encourage us to use this technique to build language independent retrieval systems. Comparing with the other participants of the AdHoc task this experiment obtained the best MAP for British Library collection and the third best for the other two. In the English and German collections the MAP results outperform the optimized MAP for the 2008 topics which is impressive and prove that the model is very strong. The French collection is the only one where the results loose significantly (~5%) for the optimized ones. We need to perform more evaluations to check this result but we also found that the problem is general to all the other participations in AdHoc track.

6 Conclusions

The results obtained support the hypotheses that using Rocchio query expansion together with a weighted model of N-Grams and a ranking scheme can be beneficial to the CLEF ad-hoc task. Applying this technique to bibliographic records using the prefix stemming instead of a sliding window to tokenize words outperforms the Porter stemming technique in most scenarios, especially when the linguistic stemmers are not appropriate. This technique can be used with different retrieval models, vector space or language modeling, because terms are scored independently. Unlike last year where our experiments resulted in poor results for both the French and German collections, this year we obtained very encouraging results. Like last year we realize that multinomial language model performs almost equivalently to vector space model in most of the situations. On other hand the multinomial language model has the advantage that we could train it very easily just by tuning the language model parameters, which was not our objective in this experiment, so we believe that language model has potential to return even better results than the vector space model.

The results obtained in this experiment support the future implementation of this model in the TEL (The European Library) search service with full text. In fact, the degraded texts resulting from the OCR of digitized works that will be provided by the TEL partners in different languages fit very well within the scope of this model.

At this time the major problem of the TEL system when using this technique is the size of the indexes that were created, reaching a total of 300 Gigabytes for 30 million pages. For future work we plan to evaluate and increase the system performance.

References

1. Porter, M.F.: An algorithm for suffix stripping. In: Sparck Jones, K., Willett, P. (eds.) Readings in Information Retrieval, pp. 313–316. Morgan Kaufmann, San Francisco (1980)
2. Hiemstra, D.: Using Language Models for Information Retrieval: Ph.D. Thesis, Centre for Telematics and Information Technology, University of Twente (2001)
3. Rocchio, J.J.: Relevance Feedback in Information Retrieval. In: The SMART Retrieval System. Experiments in Automatic Document Processing, Prentice Hall, Englewood Cliffs (1971)
4. Machado, J.: Mitra: A Metadata Aware Web Search Engine for Digital Libraries: M.Sc. Thesis, Departamento de Engenharia Informática, Technical University of Lisbon (2008)
5. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 extension to multiple weighted fields. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, CIKM 2004, Washington, D.C., USA, November 08 - 13, pp. 42–49. ACM, New York (2004)
6. Ahn, D.D., Azzopardi, L., Balog, K., Fissaha, A.S., Jijkoun, V., Kamps, J., Müller, K., de Rijke, M., Sang, E.T.K.: The University of Amsterdam at TREC 2005: Working Notes for the 2005 Text Retrieval Conference (2005)
7. Pedrosa, G., Luzio, J., Manguinhas, H., Martins, B.: DIGMAP: A service for searching and browsing old maps. In: Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2008, Pittsburgh PA, PA, USA, June 16 - 20, p. 431. ACM, New York (2008)
8. Machado, J., Martins, B., Borbinha, J.: LGTE: Lucene Extensions for Geo-Temporal Information Retrieval. In: European Conference on Information Retrieval, at Workshop on Geographic Information on Internet, Toulouse (April 2009)
9. Parapar, J., Freire, A., Barreiro, Á.: Revisiting N-gram Based Models for Retrieval in Degraded Large Collections. In: European Conference on Information Retrieval, Toulouse (April 2009)

Evaluation of Perstem: A Simple and Efficient Stemming Algorithm for Persian

Amir Hossein Jadidinejad[1], Fariborz Mahmoudi[1], and Jon Dehdari[2]

[1] Electrical and Computer Engineering Department,
Islamic Azad University, Qazvin Branch
amir@jadidi.info, mahmoudi@qiau.ac.ir
[2] Department of Linguistics
The Ohio State University
jonsafari@ling.osu.edu

Abstract. Persian is a challenging language in the field of NLP. Right-to-left orthography, complex morphology, complicated grammatical rules, and different forms of letters make it an interesting language for NLP research. In this paper we measure the effectiveness of a simple and efficient stemming algorithm, *Perstem*, on Persian information retrieval. Our experiments on the Hamshahri corpus at CLEF2009 show that the Perstem algorithm greatly improved *both* precision (*+91%*) and recall (*+43%*).

1 Introduction and Motivation

Stemmers are programs that find morphological stems of words. Stemming is a widely used method of word standardization, designed to allow the matching of morphologically related terms. Using stemmed words instead of the original words can increase overall performance of information retrieval systems. Persian is a challenging language in this field of study. *Bon* was an early Persian stemmer, proposed by Tashakori et al. [13]. Dolamic and Savoy [5] proposed using a stop word list and a light stemmer, and evaluate them with different retrieval models. Also [11], [9], [10] and [12] proposed other rule-based stemming algorithms.

Unfortunately most of the previous works evaluated using a limited subset of words [11] [4] or a limited corpus [13] [12], while the effectiveness of a stemming algorithm can be evaluated on a Persian IR system using standard and verified measures (such as precision and recall). In this paper we evaluate *Perstem* [4], a simple and efficient stemming algorithm for Persian.

2 Perstem

Perstem[1] is a stemmer and light morphological analyzer for Persian written by Jon Dehdari[2] [4]. It is written in Perl and uses a series of regular expression

[1] http://sourceforge.net/projects/perstem
[2] http://ling.ohio-state.edu/~jonsafari

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 98–101, 2010.

substitutions to separate inflectional morphemes from the stem. Input may be encoded in either UTF-8, Windows-1256, ISIRI 3342[3], HTML character entities, or romanized text. The input is then isomorphically mapped to an internal romanization format, for performance and internal consistency.

Words are first looked-up in a small hash table, and if the word matches a key, the associated value is output and no regular expression substitutions occur. This preliminary step of using a hash table serves multiple purposes. The primary purpose is to speed up treatment of the most commonly-occurring words, allowing a cached version of the morphological analysis to be output instead of subjecting a frequent word to the dozens of regular expression matches and substitutions. An example of this type of entry in the hash table is found in the first line of Table 1, `dAdnd` *dādænd* 'they gave'.

Table 1. Example hash table entries

Key	*Value*
`dAdnd`	`dAd_+nd`
`ktb`	`ktAb`
`dr`	
`sAzmAn`	`sAzmAn`

Another purpose that the hash lookup serves is to help stem *broken plural* forms of common words. Broken plurals in Persian are idiosyncratic forms of plural nouns, which were borrowed from Arabic. There are no regular ways of analyzing or stemming these words, so a lookup table is ideal for these forms. Line two of Table 1 shows a broken plural form `ktb` *kotob* 'books', resolved to the singular form `ktAb` *ketāb*.

A third purpose is to remove stopwords. These have an empty string as their associated value in the hash table, as is shown in the third line of the table, with `dr` *dær* 'in'.

The final purpose of the hash table is to correct a few high-frequency words that otherwise get stemmed incorrectly by the regular expressions. The fourth line of Table 1 shows `sAzmAn` *sāzmān* 'organization', which would have been analyzed by the regular expression substitutions as a stem `sAzm` with a plural suffix `An`.

The vast majority of the input words do not match any key in the hash table[4], and instead are analyzed by a series of regular expression substitutions. These substitutions identify inflectional prefixes and place a `+_` between the prefix and the stem, and likewise identify inflectional suffixes and place a `_+` between the stem and the suffix. For example, a word like `nmi-xurdndC` *nemi-xordændeš* 'they were not eating it' would first get analyzed as `n+_mi-+_xur_+d_+nd_+C`.

[3] http://www.isiri.org/std/3342.htm

[4] More than 70% of tokens in running news text do not match. The hash table has about 130 entries.

Then stemming is simply a matter of deleting all prefixes (X+_) and all suffixes (_+Y). So the previous example would get stemmed to `xur` *xor* 'eat'.

Perstem currently has about 50 regular expression substitution rules. It has shown very good levels of accuracy and speed in previous experiments, correctly analyzing 97% of the words in the test set [4]. Most of the errors involve words ending in the letter *ye*, which can be a derivational attributive suffix, an inflectional specific indefinite suffix, or may simply be part of the stem. The stemmer has a command-line argument `--recall` that enables a few extra regular expressions for highly-ambiguous analyses, such as words ending in the letter *ye*. Enabling this argument will usually increase recall at the expense of precision. The stemmer can process 15,000 words *per second* on a basic desktop computer.

3 Experiments

In order to investigate the effectiveness of Perstem, we perform two different experiments using the same retrieval model. First, the original Hamshahri corpus [3] is indexed and retrieved using the Indri search engine [8]; this experiment is called "Baseline". After that, Perstem [4] analyzes all documents and queries in the Hamshahri corpus and creates a new stemmed corpus and queries. The same indexing and retrieval method [8] is applied on this new stemmed collection. This run is called "Perstem". Table 3 compares the effectiveness of these experiments[5] at CLEF2009 [6]. Compared to the previous experiments [13] [7] [5] [2] that improved recall but sometimes hurt precision, our results are interesting in that it greatly improves *both* recall and precision.

Table 2. Performance comparison for the monolingual Persian track at CLEF2009 [6]. "Baseline" is a retrieval experiment using the Indri retrieval engine [8] without stemming. "Perstem" is a similar experiment on the stemmed corpus and queries.

Run Name	Relevant-Retrieved	Recall	MAP	P@5,10,15,20	NDCG	R-PREC
Baseline	2670/4464	0.5981	0.1964	0.36,0.36,0.35,0.33	0.4649	0.2345
Perstem	3820/4464	0.8557	0.3762	0.58,0.56,0.55,0.53	0.7089	0.4033
	+43%	**+43%**	**+91%**	**+60%**	**+52%**	**+72%**

A query-by-query comparison confirms the findings discussed in section 2, that while Perstem is effective for most words, using the `--recall` argument (which we did) hurt precision when dealing with many nouns ending in the letter *ye*. For example, in query 10.2452/649-AH, *xātæmi* (5$^{\text{th}}$ President of Iran) is stemmed to *xātæm* (marquetry), which hurt the precision as much as *73%*. Future work could include further analyzing these types of errors—frequent, ambiguous words—and selectively adding some to the hash table. We are also interested in performing similar experiments without the `--recall` flag enabled.

[5] DOI: 10.2415/AH-PERSIAN-MONO-FA-CLEF2009.QAZVINIAU.IAUPERFA3-4.

References

1. Agirre, E., Di Nunzio, G.M., Ferro, N., Mandl, T., Peters, C.: CLEF 2008 Ad hoc track overview. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 15–37. Springer, Heidelberg (2009)
2. AleAhmad, A., Kamalloo, E., Zareh, A., Rahgozar, M., Oroumchian, F.: Cross Language Experiments at Persian@CLEF 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 105–112. Springer, Heidelberg (2009)
3. AleAhmad, A., Amiri, H., Darrudi, E., Rahgozar, M., Oroumchian, F.: Hamshahri: A standard Persian text collection. Knowledge-Based Systems 22(5), 382–387 (2009)
4. Dehdari, J., Lonsdale, D.: A link grammar parser for Persian. Aspects of Iranian Linguistics, vol. 1. Cambridge Scholars Press (2008)
5. Dolamic, L., Savoy, J.: Persian Language, Is Stemming Efficient? In: 20th International Workshop on Database and Expert Systems Application, Linz, Austria, pp. 388–392 (2009)
6. Ferro, N., Peters, C.: CLEF 2009 Ad Hoc Track Overview: TEL & Persian Tasks. In: Workshop on Cross-Language Information Retrieval and Evaluation, Corfu, Greece (2009)
7. Karimpour, R., Ghorbani, A., Pishdad, A., Mohtarami, M., AleAhmad, A.: Using Part of Speech tagging in Persian Information Retrieval. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, Springer, Heidelberg (2009)
8. Metzler, D., Croft, W.B.: Combining the Language Model and Inference Network Approaches to Retrieval. Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval 40(5), 735–750 (2004)
9. Mokhtaripour, A., Jahanpour, S.: Introduction to a new Farsi stemmer. In: 15th ACM International Conference on Information and Knowledge Management. ACM, USA (2006)
10. Shahbazi, H., Mokhtaripour, A., Dalvi, M., Tork Ladani, B.: A New Approach for Scoring Relevant Documents by Applying a Farsi Stemming Method in Persian Web Search Engines. In: 13th International CSI Computer Conference, Kish Island, Iran, pp. 745–748 (2008)
11. Sharifloo, A., Shamsfard, M.: A Bottom Up approach to Persian Stemming. In: Third International Joint Conference on Natural Language Processing. ACL, India (2008)
12. Taghva, K., Beckley, R., Sadeh, M.: A Stemming Algorithm for the Farsi Language. In: International Conference on Information Technology: Coding and Computing. IEEE Computer Society, USA (2005)
13. Tashakori, M., Meybodi, M., Oroumchian, F.: Bon: First Persian Stemmer. In: First Eurasia Conference on Advances in Information and Communication Technology, Tehran, Iran (2002)

Ad Hoc Retrieval with the Persian Language

Ljiljana Dolamic and Jacques Savoy

Computer Science Department, University of Neuchatel,
Rue Emile Argand 11, 2009 Neuchâtel, Switzerland
{Ljiljana.Dolamic,Jacques.Savoy}@unine.ch

Abstract. This paper describes our participation to the Persian *ad hoc* search during the CLEF 2009 evaluation campaign. In this task, we suggest using a light suffix-stripping algorithm for the Farsi (or Persian) language. The evaluations based on different probabilistic models demonstrated that our stemming approach performs better than a stemmer removing only the plural suffixes, or statistically better than an approach ignoring the stemming stage (around +4.5%) or a n-gram approach (around +4.7%). The use of a blind query expansion may significantly improve the retrieval effectiveness (between +7% to +11%). Combining different indexing and search strategies may further enhance the MAP (around +4.4%).

1 Introduction

Our participation to the CLEF 2009 evaluation campaign was motivated by our objective to design and evaluate indexing and search strategies for other languages than English studied since 1960. In fact, other natural languages may reveal different linguistic constructions having an impact on the retrieval effectiveness. For some languages (e.g., Chinese, Japanese [1]), word segmentation is not an easy task, while for others (e.g., German), the use of different compound constructions to express the same concept or idea may hurt the retrieval quality [2]. The presence of numerous inflectional suffixes (e.g., Hungarian [3], Finnish), even for names (e.g., Czech [4], Russian [5]) as well as numerous derivational suffixes must be taken into account for an effective retrieval.

In this context, the Persian language is member of the Indo-European family written using Arabic letters. The underlying morphology [6] is slightly more complex than the English one but we cannot qualify it as hard compared to some languages such as Turkish or Finnish.

The rest of this paper is organized as follows. The next section describes the main characteristics of the Persian morphology and presents an overview of the test-collection. Section 3 exposes briefly the various IR models used in our evaluation. The evaluation of the different indexing and search models are described and analyzed in Section 4 follows by the description of our official results. Our main findings are regrouped in the last section.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 102–109, 2010.

2 Farsi (Persian) Language and Test-Collection

The Persian language, belonging to the Indo-Aryan language family is written using 28 Arabic letters, with additional 4 letters (پ چ ژ گ) being added to express sounds not present in classical Arabic. The morphology of this language is based on various suffixes used to indicate the plural, the accusative or genitive cases as well as other suffixes (or prefixes) are employed to derive new words. The plurals in the Persian are formed by means of two suffixes, namely ان for animate (پدر, father, پدران, fathers) and ها for inanimate (گل, flower, گلها, flowers) nouns, while the plural of Arabic nouns in this language is formed according to Arabic grammar rules (e.g., ات or ین for "sound" plurals). Moreover, even though this language does not have the definite article in the strict sense, it can be said that the relative suffix ی (کتابی که, the book which) and suffix ه (پسره, the son, informal writing) perform this function.

The suggested "light" stemmer[1] removes the above mentioned suffixes with addition of certain number of possessive and comparative suffixes. It is clearly less aggressive than, for example, the Porter's stemmer [7] used in the English language. The second stemmer we proposed, denoted "plural", detects and removes only the plural suffixes from Persian nouns together with any suffix that might follow them. This stemmer is similar to the English S-stemmer [8]. As a stemming strategy we may also consider using a morphological analysis [9]. Recent research demonstrates however that using a morphological analysis, a light or a more aggressive stemming approaches tend to produce statistically similar performance, at least for the English language [10].

To evaluate these various stemming approaches we will use the Persian test-collection composed of newspaper articles extracted from *Hamshahri* (covering years 1996 to 2002). This corpus is the same one made available during the CLEF 2008 campaign containing 166,477 documents. In mean, we can find 202 terms per document (after stopword removal). The available documents do not have any logical structure and are composed of a few paragraphs. During the indexing process, we have found 324,028 distinct terms.

The collection contains 50 new topics (numbered from Topic #600 to Topic #650) having total of 4,464 relevant items, with mean of 89.28 relevant items per query (median 81.5, standard deviation 55.63). The Topic #610 ("Benefits of Copyright Laws") has the smallest number of relevant items (e.g., 8) while the largest number of relevant items (e.g., 266) was found for the Topic #649 ("Khatami Government Oil Crisis").

3 IR Models

In order to analyze the retrieval effectiveness under different conditions, we adopted various retrieval models for weighting the terms included in queries and documents. To be able to compare the different models and analyze their

[1] Freely available at http://www.unine.ch/info/clef/

relative merit, we first used a classical *tf idf* model. We would thus take into account the occurrence frequency of the term t_j in the document D_i (or tf_{ij}) as well as its inverse document frequency ($idf_j = \ln(\frac{n}{df_j})$ with n the number of documents in the corpus, and df_j the number of documents in which t_j occurs). Furthermore we normalized each indexing weight using the cosine normalization.

To define the similarity between a document surrogate and the query, we compute the inner product as given by Equation 1.

$$score(D_i, Q) = \sum_{t_j \in Q} w_{ij} \cdot w_{Qj} \tag{1}$$

where w_{ij} represents the weight assigned to the term t_j in the document D_i and w_{Qj} the weight assigned to t_j in the query Q.

As other IR model, we implemented several probabilistic approaches. As a first probabilistic approach, we implemented the Okapi model (BM25) [11] evaluating the document score by applying following formula:

$$score(D_i, Q) = \sum_{t_j \in Q} qtf_j \cdot \log\left[\frac{n - df_j}{df_j}\right] \cdot \frac{(k_1 + 1) \cdot tf_{ij}}{K + tf_{ij}} \tag{2}$$

with $K = k_1 \cdot ((1 - b) + b \cdot \frac{l_i}{avdl})$ where qtf_j denotes the frequency of term t_j in the query Q, and l_i the length of the document D_i. The average document length is given by $avdl$ while b (=0.75) and k_1 (=1.2) are constants.

As second probabilistic approach, we implemented several models issued from the *Divergence from Randomness* (DFR) paradigm [12]. In this framework, two information measures are combined to compute the weight w_{ij} attached to the term t_j in the document D_i as shown in Equation 3.

$$w_{ij} = Inf^1_{ij} \cdot Inf^2_{ij} = -\log_2(Prob^1_{ij}(tf_{ij})) \cdot (1 - Prob^2_{ij}(tf_{ij})) \tag{3}$$

As a first model, we implemented the DFR-PL2 scheme, defined by Equation 4 and 5.

$$Prob^1_{ij} = \frac{e^{-\lambda_j} \cdot \lambda_j^{tfn_{ij}}}{tfn_{ij}!} \tag{4}$$

$$Prob^2_{ij} = \frac{tfn_{ij}}{tfn_{ij} + 1} \tag{5}$$

with $\lambda_j = \frac{tc_j}{n}$ and $tfn_{ij} = tf_{ij} \cdot \log_2(1 + \frac{c \cdot mean\_dl}{l_i})$ where tc_j represents the number of occurrences of term t_j in the collection. The constants c and $mean\_dl$ (average document length) are fixed according to the underlying collection.

As second DFR model, we implemented the DFR-In_eC2 model defined by following equations, with $n_e = n \cdot (1 - (\frac{n-1}{n})^{tc_j})$.

$$Inf^1_{ij} = tfn_{ij} \cdot \log_2\left[\frac{n + 1}{n_e + 0.5}\right] \tag{6}$$

$$Prob^2_{ij} = 1 - \frac{tc_j + 1}{df_j \cdot (tfn_{ij} + 1)} \tag{7}$$

Finally we also used a non-parametric probabilistic model based on a statistical language model. In this study we adopted a model proposed by Hiemstra [13] combining an estimate based on document ($P(t_j|D_i)$) and on corpus ($P(t_j|C)$) and defined by following equation:

$$P(D_i|Q) = P(D_i) \cdot \prod_{t_j \in Q} [\lambda_j \cdot P(t_j|D_i) + (1 - \lambda_j) \cdot P(t_j|C)] \quad (8)$$

with $P(t_j|D_i) = \frac{tf_{ij}}{l_i}$ and $P(t_j|C) = \frac{df_j}{lc}$ with $lc = \sum_k df_k$ where λ_j is a smoothing factor, and lc an estimate of the size of the corpus C. In our experiments λ_j is constant (fixed at 0.35) for all indexing terms t_j.

4 Evaluation

To measure retrieval performance we used the mean average precision (MAP) obtained from 50 queries. The best performance obtained under a given condition is shown in bold type in the following tables. In order to statistically determine whether or not a given search strategy would be better than another, we applied the bootstrap methodology [14] based on two-sided non-parametric test ($\alpha = 5\%$). In all experiments presented in this paper our stoplist[2] for the Persian language containing 884 terms has been used.

Table 1. MAP of Various Indexing Strategies and IR models (T query)

Query T	Mean Average Precision (MAP)					
Stemmer	none	plural	light	perstem	5-gram	trunc-4
Okapi	0.3687†	0.3746†	0.3894†	0.3788†	0.3712†	0.3954†
DFR-PL2	**0.3765**	**0.3838**	**0.3983**	0.3879	0.3682†	**0.4054**
DFR-In_eC2	0.3762	0.3830	0.3952	**0.3886**	**0.3842**	0.4016
LM	0.3403†	0.3464†	0.3559†	0.3471†	0.3404†	0.3546†
tf idf	0.2521†	0.2632†	0.2521†	0.2575†	0.2441†	0.2555†
Mean	0.3428	0.3502	0.3582	0.3520	0.3416	0.3625
% over "none"		+2.17%	+4.50%	+2.69%	-0.33%	+5.76%

Table 1 shows the MAP achieved by five IR models as well as different indexing strategies with the short query formulation. The second column in Table 1 (marked "none") depicts the performance obtained by the word-based indexing strategy without stemming, followed by the MAP achieved by our two stemmers, namely "plural" and "light". In the column marked "perstem" the results obtained using publicly available stemmer and morphological analyzer for the Persian language[3] are given. This stemmer is based on numerous regular expressions to remove the corresponding suffixes. Finally the last two columns

[2] Freely available at http://www.unine.ch/info/clef/

[3] http://sourceforge.net/projects/perstem/

depict the performance of two language independent indexing strategies, namely 5-gram and trunc-4 [15]. With the n-gram approach, each word is represented with overlapping sequences of n characters (e.g., from "computer", we obtain "compu", "omput", "mpute", and "puter"). With trunc-4, we retain only the first n letter and, for example, the word "computer" will produce "comp". The values of 5 and 4 are selected to obtain the best possible performance.

It can be seen from this table that the best performing models for all indexing strategies are the models derived from the DFR paradigm (marked bold in the table). To verify whether this retrieval performance is significantly better than the other IR models, we have applied our statistical test. In Table 1, we have added the symbol "†" after MAP values showing a statistically significant differences with the best performance. Clearly the best IR model is always significantly better than the classical *tf idf* vector-space model or than the LM approach. If the Okapi model performs always at a lower performance level, the differences are usually not statistically significant.

When analyzing the various stemming approaches, the best performing indexing strategy seems to be the "light" stemming approach. An an exception we can mention the *tf idf* IR model for which the best performance was obtained by "plural" indexing approach (0.2632). From data shown in Table 1, even if the "light" stemmer is the best approach, the performance differences are usually significant only when compared to an approach ignoring the stemming stage. Finally, the performance differences between both language-independent approaches (n-gram and trunc-n) and our "light" stemming are never statistically significant.

We performed a query-by-query comparison to understand the effect of stemming concentrating on DFR-PL2, the best performing IR model. Analyzing Topic #630 ("Iranian Traditional Celebrations") we come across almost full range of reasons for better performance of the light stemming resulting in MAP 0.3808 compared to 0.1458 achieved by "none" or 0.2042 by trunc-4. While the topic title contains the adjectives ایرانی (Iranian) and سنتی (traditional), the relevant documents contain also ایران (Iran), سنت (tradition), ستهای (traditions) being conflated into the same respective indexing term by our "light" stemmer, but not when ignoring the stemming stage. Topic contains also the plural form of the noun جشنهای (the celebrations) while جشن (celebration) and جشنها (celebrations) are also found in the relevant documents. With the trunc-4 scheme, the resulting indexing term is composed of three letters (the stem "celebration") and one letter of the suffix. Thus it is not possible to conflate the two forms "celebration" (3 letters) and "celebrations" (5 letters) under the same entry.

Table 2 shows the MAP obtained using two different indexing strategies, namely "none" and "light" over five IR models with three query formulations (short or T, medium or TD and the longest form or TDN). It can be seen that augmenting the query size improves the MAP over T query formulation by +8% in average for TD queries and +15% for TDN queries. Moreover, the performance difference that are statistically significant over the T query formulation are shown with the symbol "‡".

Table 2. MAP of Various IR Models and Query Formulations

	Mean Average Precision					
Query	T	TD	TDN	T	TD	TDN
Stemmer	none	none	none	light	light	light
Okapi	0.3687	0.3960‡	0.4233‡	0.3894	0.4169‡	0.4395‡
DFR-PL2	**0.3765**	**0.4057‡**	**0.4326‡**	0.3983	**0.4247‡**	**0.4521‡**
DFR-In_eC2	0.3762	0.4051‡	0.4284‡	**0.4226**	0.4226	0.4417‡
LM	0.3403	0.3727‡	0.4078‡	0.3559	0.3867‡	0.4268‡
tf idf	0.2521	0.2721	0.2990	0.2521	0.2687	0.2928‡
mean	0.3428	0.3703	0.3982	0.3582	0.3839	0.4106
% over T		+8.0%	+16.2%		+7.2%	+14.6%

Upon inspection of obtained results, we have found that the pseudo-relevance feedback can be useful to enhance retrieval effectiveness. Table 3 depicts MAP obtained by using Rocchio's approach (denoted "Roc") [16] whereby the system was allowed to add m terms (m varies from 20 to 150) extracted from the k best ranked documents (for $k = 5$ to 10) from the original query results. The MAP enhancement spans from +2.4% (light, Okapi, 0.4169 vs. 0.4267) to +11.1% (light, DFR-PL2, 0.4247 vs. 0.4718). We have also applied another *idf*-based query expansion model [17] in our official runs (see Table 4).

Table 3. MAP using Rocchio's Blind-Query Expansion

	Mean Average Precision			
Query	TD	TD	TD	TD
Index	light	light	5-gram	5-gram
IR Model/MAP	Okapi 0.4169	DFR-PL2 0.4247	Okapi 0.3968	DFR-PL2 0.3961
PRF Rocchio	5/20 0.4306	5/20 0.4621	5/50 0.4164	5/50 0.4164
k doc./m terms	5/70 **0.4480**	5/70 0.4620	5/150 0.4238	5/150 **0.4238**
	10/20 0.4267	10/20 **0.4718**	10/50 0.4173	10/50 0.4173
	10/70 0.4441	10/70 0.4700	10/150 **0.4273**	10/150 0.4169

5 Official Results

Table 4 gives description and results of the four official runs submitted to the CLEF 2009 Persian *ad hoc* track. Each run is based on a data fusion scheme combining several single runs using different IR models (DFR, Okapi, and language model (LM)), indexing strategies (word with and without stemming or 5-gram), query expansion strategies (Rocchio, *idf*-based or none) and query formulation (T, TD, and TDN). The fusion was performed for all four runs using our Z-score operator [18]. In all cases we can see that combining different models, indexing and search strategies using Z-score approach improves clearly the retrieval effectiveness. For example, using the short query formulation (T), the best single

IR model achieved a MAP value of 0.4197, while after applying our data fusion operator, we obtained a MAP value of 0.4380, a relative improvement of +4.3%. In these different combinations, we however did not use our "light" stemmer showing a relatively hight retrieval effectiveness as depicted in Table 1.

Table 4. Description and MAP of Official Persian Runs

Run name	Query	Index	Model	Query exp.	MAP	Comb.MAP
UniNEpe1	T	word	PL2	none	0.3765	**0.4380**
	T	5-gram	LM	idf 10 docs/50 terms	0.3726	
	T	plural	Okapi	Roc 10 docs/70 terms	0.4197	
UniNEpe2	TD	5-gram	In_eC2	none	0.4113	0.4593
	TD	word	PL2	none	0.4057	
	TD	plural	Okapi	Roc 5 docs/70 terms	0.4311	
	TD	word	PL2	idf 10 docs/50 terms	0.4466	
UniNEpe3	TD	word	Okapi	Roc 5 docs/50 terms	0.4228	**0.4663**
	TD	plural	Okapi	Roc 5 docs/70 terms	0.4311	
	TD	perstem	PB2	idf 10 docs/50 terms	0.4462	
UniNEpe4	TDN	word	LM	Roc 10 docs/50 terms	0.4709	**0.4937**
	TDN	plural	Okapi	Roc 5 docs/70 terms	0.4432	
	TDN	perstem	PL2	Roc 10 docs/20 terms	0.4769	

6 Conclusion

From our past experiences in various evaluation campaigns, the results achieved in this track confirm the retrieval effectiveness of the *Divergence from Randomness* probabilistic model family. In particular the DFR-PL2 or the DFR-In_eC2 implementation tends to produce high MAP when facing different test-collections written in different languages. We can also confirm that using our Z-score operator to combine different indexing and search strategies tends to improve the resulting retrieval effectiveness.

In this Persian *ad hoc* task, we notice three main differences between results achieved last year and those obtained this year. First, using short (title-only or T) query formulation, we achieved the best results in 2008. This is the contrary this year with results based on TDN topic formulation depicting the best MAP (see Table 2). Second, unlike last year, the use of our stemmers was effective this year, and particularly the "light" stemming approach (see Table 1). As language-independent approach, we can mention that the trunc-n indexing scheme is also effective for the Persian language. Third, applying a pseudo-relevance feedback enhance the retrieval effectiveness of the proposed ranked list (see Table 3). For the moment, we do not have found a pertinent explanation to such difference between the two years. However, during both evaluation campaigns we found that a word-based indexing scheme using our "light" stemmer tends to perform better than the n-gram scheme.

Acknowledgments. The authors would like to also thank the CLEF-2009 organizers for their efforts in developing this test-collection. This research was supported in part by the Swiss National Science Foundation (Grant #200021-113273).

References

1. Savoy, J.: Comparative Study of Monolingual and Multilingual Search Models for Use with Asian Languages. ACM Transactions on Asian Languages Information Processing 4, 163–189 (2005)
2. Braschler, M., Ripplinger, B.: How Effective is Stemming and Decompounding for German Text Retrieval? IR Journal 7, 291–316 (2004)
3. Savoy, J.: Searching Strategies for the Hungarian Language. Information Processing & Management 44, 310–324 (2008)
4. Dolamic, L., Savoy, J.: Indexing and Stemming Approaches for the Czech Language. Information Processing & Management 45, 714–720 (2009)
5. Dolamic, L., Savoy, J.: Indexing and Searching Strategies for the Russian Language. Journal of the American Society for Information Sciences and Technology 60, 2540–2547 (2009)
6. Elwell-Sutton, L.P.: Elementary Persian Grammar. Cambridge University Press, Cambridge (1999)
7. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14, 130–137 (1980)
8. Harman, D.K.: How Effective is Suffixing? Journal of the American Society for Information Science 42, 7–15 (1991)
9. Miangah, T.M.: Automatic Lemmatization of Persian Words. Journal of Quantitative Linguistics 13, 1–15 (2006)
10. Fautsch, C., Savoy, J.: Algorithmic Stemmers or Morphological Analysis: An Evaluation. Journal of the American Society for Information Sciences and Technology 60, 1616–1624 (2009)
11. Robertson, S.E., Walker, S., Beaulieu, M.: Experimentation as a Way of Life: Okapi at TREC. Information Processing & Management 36, 95–108 (2002)
12. Amati, G., van Rijsbergen, C.J.: Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness. ACM Transactions on Information Systems 20, 357–389 (2002)
13. Hiemstra, D.: Using Language Models for IR. Ph.D. Thesis (2000)
14. Savoy, J.: Statistical Inference in Retrieval Effectiveness Evaluation. Information Processing & Management 33, 495–512 (1997)
15. McNamee, P., Nicholas, C., Mayfield, J.: Addressing Morphological Variation in Alphabetic Languages. In: Proceedings ACM - SIGIR, 75–82 (2009)
16. Buckley, C., Singhal, A., Mitra, M., Salton, G.: New Retrieval Approaches Using SMART. In: Proceedings TREC-4, Gaithersburg, pp. 25–48 (1996)
17. Abdou, S., Savoy, J.: Searching in Medline: Stemming, Query Expansion, and Manual Indexing Evaluation. Information Processing & Management 44, 781–789 (2008)
18. Savoy, J.: Combining Multiple Strategies for Effective Monolingual and Cross-Lingual Retrieval. IR Journal 7, 121–148 (2004)

Ad Hoc Information Retrieval for Persian

AmirHossein Habibian, Abolfazl AleAhmad, and Azadeh Shakery

Database Research Group, School of Electrical and Computer Engineering
University of Tehran
{Habibian,AleAhmad,Shakery}@ut.ac.ir

Abstract. In this paper we present an introduction to the Persian language and its morphology, and describe available resources for Persian text processing. We then propose and evaluate an information retrieval model, a variation of the vector space model which uses the relations existing between query terms. Our experiments on the Hamshahri collection show that the proposed model has better precision for top ranked documents in comparison with some popular IR models.

1 Introduction

While information retrieval (IR) has been exhaustively explored for many languages, this is not the case for Persian. The Persian language has some common characteristics with Arabic, namely, it is written from right to left and contains 32 letters, 26 of which are common to Arabic. However, the vocabulary and grammar of Persian is different. This implies that information retrieval for this language needs some special considerations.

Previously, the lack of a standard Persian collection was a hindrance when studying the language but, thanks to our collaboration with CLEF, at the Database Research Group in the University of Tehran, we have been able to create a large text collection, known as Hamshahri, in order to investigate Persian IR. The collection contains more than 160,000 Persian text documents from the Hamshahri newspaper. As a result of using the collection in the CLEF 2008 and 2009 campaigns, 100 standard topics with their relevance judgment were created. However, a goal of our participation in these two CLEF campaigns has not only been to build a standard text collection for Persian IR but also to develop much needed language processing algorithms for Persian, such as stemmers, tokenizers, etc.

Recently we further developed the collection, building Hamshahri2 with some new features. The new version is twice as large as the version that we used in CLEF. Table 1 compares some characteristics of the two collections. In addition, Hamshahri2, includes more than148,000 images together with the news articles, for a further 1.9 GB. Hamshahri2 is thus a useful resource for people studying algorithms for image retrieval[1].

[1] For more info about the Hamshahri collection, the reader is referred to [7] or the collection's web site: http://ece.ut.ac.ir/DBRG/Hamshahri

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 110–119, 2010.

Table 1. Comparison between old and new versions of the Hamshahri collection

Attribute	Hamshahri 1	Hamshahri2
Collection size (Unicode)	611 MB	1.4 GB
Documents format	XML	XML
Documents date span	1996-2002	1996-2007
No. of documents	166,000+	318,000+
No. of main categories	9	9
No. of Topics	100 (2×50)	50

This paper is composed of two main parts: in Section 2, we briefly describe the Persian morphology, the special characteristics of the language that should be considered when designing an information retrieval system for Persian, and the resources and tools available for Persian text processing; in Section 3, we propose an IR model which is an extension of the vector-space model and tries to capture the relationship between query words. We report our experimental results with this model on the CLEF 2009 Persian test collections. Section 4 provides some concluding remarks. It should be noted that we were not able to submit our results at CLEF 2009, so the experiments reported in Section 3 are all post-campaign runs.

2 The Persian Language and Morphology

In this section, we present a brief description of Persian morphology. Our intention is not an exhaustive description but rather to give the reader a sense of the challenges he/she may encounter when dealing with Persian information retrieval. More details of Persian morphology can be found in [1, 7, 8, 9]. We will also introduce resources and tools available for Persian text processing.

2.1 Persian Morphology

Persian or Farsi, an Indo-European language, is the official language of Iran, Afghanistan and Tajikistan and is widely used as a second language in some other countries. As an Indo-European language, Persian has some common features with English. For example they both use prefixes and suffixes to form new words. But the writing system is quite different. Persian is written from right to left and the script is an extended version of the Arabic script.

There are some certain features in Persian which can be sources of ambiguities. Short vowels are usually not written in Farsi, giving rise to words with the same surface form but quite different meanings. For instance, the word "مردم" can either mean *"mardom"* ("people") or *"mordam"* ("I died") depending on the context. Even worse, the word "کرم" can have several pronunciations and meanings: *"kerm"* ("worm"), *"korom"* ("chrome"), *"karam"* ("generosity"), *"karam"* ("I am deaf") and *"kerem"* ("cream"). Another main source of ambiguity is discontinuity in the structure of words: some prefixes/suffixes are always bound, while others can appear with a space and be free.

In Persian, many words are created from the imperative form of verbs and thus understanding the imperative form of words is very important. For example, the stem of

the word "خواننده" ("Reader" in English) is "خوان" ("Read" in English), which is obtained by removing the suffix "نده" from the word. However, as Persian contains irregular infinitives, obtaining the imperative forms is not an easy task. Regular infinitives end with the suffix "دن" and imperative forms can be obtained by eliminating the last 2 or 3 letters of words (e.g. "پرسیدن" ("to ask" in English) is a regular infinitive and "پرس" ("ask" in English) is the imperative form). But there is no special rule for obtaining imperatives from irregular infinitives. For example "رفتن" ("to go") is obtained from "رو" ("go") and "کردن" ("to do") is obtained from "کن" ("do"). The imperative form of irregular infinitives is formed based on their usage and pronunciation. The two prefix letters "ب" and "ن" are used to form positive and negative verb forms respectively. For example "برو" ("go") and "نرو" ("do not go") are the positive and negative forms of "رو" ("to go") respectively.

If we obtain the imperative form of a verb (that is a tense root), then we will be able to create different variations of the verb. For example, "خور" is the imperative form of the infinitive "خوردن", then if we append the person suffix "م" to the end of the imperative and the tense prefix "می" to the beginning, we will have "می‌خورم" ("I eat" in English) which is the present simple tense of the verb. The past tense of verbs is obtained by eliminating "ن" from infinitives. For example consider the infinitive "خوردن" again, after eliminating "ن" we have "خورد", which is the past form of the verb, and if we add the person suffix "م" it becomes "خوردم", which means "I ate" in English.

A number of verbs in Farsi are light verbs which can be used to form other verbs. "کردن" ("do, make"), "دادن"("give"), "خوردن"("eat") and "زدن"("hit, strike") are examples of light verbs. These verbs, combined with a noun, an adjective or a preposition can formother verbs in which the original meaning of the light verb maybe completely lost. For example, the word "سرما" ("cold weather") combined with "خوردن"("eat") will result "سرما خوردن"("to catch cold").

Persian has no gender distinctions and no agreement between noun and modifiers which makes it easier in some respects to many other languages. But on the other hand, it is very lenient when forming complex words. For example, many prefixes and suffixes can be combined with a stem, ending up in a word corresponding to a sentence in English. For example the word "کوچکترینهایشانند", composed of a stem followed by different suffixes, corresponds to the sentence "They are the smallest ones of them" in English.

One difficult part of the Persian morphology is its plural forms. In order to create the plural form of Persian nouns, different suffixes could be added to the noun based on the noun itself. Some suffixes are "ان", "ها" and "ات", "ها" is the most frequently used. For example, the plural form of the noun "کیف" (meaning "bag" or "enjoyment" based on its pronunciation) is "کیف ها" ("bags" or "enjoyments"), or "کیفها" if we remove the space between the word and the suffix which is common in Persian writing. The plural form of "مشکل" (a word imported from Arabic that means "problem") on the other hand is either "مشکلات" or "مشکل ها" ("problems") concatenating "ات" or "ها" to the word respectively. Note that we cannot use the suffix "ات" for the previous noun, namely "کیفات" is wrong. There are also plural forms that are borrowed from Arabic, which do not follow the rule of concatenating suffixes. For example, the plural form of "کتاب"("book") is "کتب"("books"). These forms usually do not undergo morphological analysis.

Persian also borrows some words from Arabic, as well as the Arabic rule-pattern system of morphology. For example the word "شعر" (“poem”) is borrowed along with its plural and participial forms: "اشعار" (“poems”), "شاعر" (“poet”), "شعرا"(“poets”) and "مشاعره" (“poetical contest”). These borrowed words are analyzed morphologically in some systems and are left without further analysis in others.

As mentioned earlier, our discussion here is far from complete. The interested reader can find more details about Persian morphology in many books that are written on this subject.

2.2 Resources and Tools

In addition to the Hamshahri collection [6] which is used in this study, there are some other resources that are useful for Persian text processing and retrieval. The Bijankhan collection is a manually tagged corpus containing nearly 2.6 million words and a tag set of 550 POS tags [10]. FarsNet is another language resource for Persian that contains lexical, syntactic and semantic knowledge for more than 15000 words [11]. A Persian spell checker and an English-Persian parallel corpus containing 612,000 bilingual sentences have also been prepared at the NLP lab of the University of Tehran [12]. The Shiraz machine translation is an open source system created at the University of New Mexico [13]. A light Persian stemmer has been developed at the University of Neuchatel [14].

3 A Method for Considering Term Relations in Text Retrieval

Sentences are composed of some terms assembled together using some language specific rules. A group of terms may represent a new concept which differs from concepts of its individual terms. For example the definition of “world cup”, an international football competition, differs from the definitions of “world” and “cup” separately. We call this meaningful combination of terms, a phrase. In Persian, like other languages, phrases are commonly used in sentences and detecting them in documents can help retrieval models to avoid retrieving some non-relevant documents.

In this section we propose a model that detects meaningful phrases in the query and retrieves relevant documents according to the detected phrases. To examine the effect of considering phrases in Persian text retrieval, we evaluate the model on the Hamshahri test collection with the query sets of CLEF 2008 and 2009. We also compare the performance of this model with some more commonly used information retrieval models for Persian text retrieval.

3.1 The Proposed Model

In the vector space model, each document is considered as a vector of terms. Elements of this vector are weights which are calculated on the basis of document term frequency. These weights are calculated separately for each term and the information which exists between terms relations is ignored. The information provided by the relations between terms can completely change the meaning of the constituent terms.

In our proposed model, we try to exploit the relations existing between query terms to make a vector of query phrases instead of terms. The main idea of the proposed

model is to search for the meaningful relations existing between query terms. This search is performed exhaustively in a set that contains the entire combinations of query terms. We introduce some criteria to measure how meaning changes when concatenating terms and thus detect important phrases. We use these criteria to weight the elements of the query phrase vector. Documents are then retrieved by an extension of the vector space model, which has been modified to handle query phrase vectors instead of query term vectors. Throughout this paper, we call our proposed model: Phrase Based Vector Space (PBVS).

In the remaining parts of this section, we introduce the two main steps of our proposed model: making query phrase vectors, and modifying the vector space model on the basis of phrase vectors.

3.1.1 Making Query Phrase Vector

Before describing our proposed model, we give a more precise definition of a query phrase. Consider each query as a set of query terms. After elimination of stop words, we name this set as "S" and each of 2^n-1 non empty subsets of S as a "phrase". In this article we use P_i to refer to the i^{th} phrase. In our work, we deal with phrases just like terms and define document frequency and term frequency for them.

Regarding the above definition, each phrase not only consists of some terms, but also preserve the relation that exists between them. We introduce some criteria to find the phrases which are more important and contain meaningful relations between their terms.

Criterion 1: This criterion is used to determine the specificity of a phrase. For this purpose we use the following formula:

$$\text{Criterion1 (Pi)} = \varepsilon + (1 - \varepsilon) \quad *\left(1 - \frac{\text{Max}_{t \in Pi}\{\text{idf(t)}\}}{\text{idf(Pi)}}\right)$$

As discussed in the next section, the document frequency of a phrase is defined as the number of documents which contain all the phrase terms. This formula measures the effect of concatenating terms of a phrase in causing the phrase to become more specific. We use *idf* to measure the specificity of the phrase and its terms. A higher ratio of phrase specificity to term specificity indicates that there may be some kind of relation between the phrase terms that make it specific and that the phrase can be more effective in document retrieval. This criterion is formulated as above to lie between ε and 1.

Criterion 2: This criterion is used to measure the portion of the information need that the phrase contains. This measurement should be defined to be maximized for the phrase that contains all the terms of the query and it should not decrease as long as it contains important keywords of the query. We define this criterion as below:

$$\text{Criterion2 (Pi)} = \frac{\sum_{t \in Pi} \text{idf(t)}}{\sum_{t \in \text{Query}} \text{idf(t)}}$$

The query phrase vector for a query with "n" non-stop word terms will be a vector with 2^n-1 non-zero elements. Each element of this vector is a phrase and we assign

some weighting to it based on the product of these criteria. It is clear that the phrase vector contains all of the query terms and can be presumed to be a superset of query terms vector.

3.1.2 Phrase-Based Vector Space

So far we have discussed how to make query phrase vector. We defined query phrase vectors as a vector in which the only elements that have non-zero weightings are 2^n-1 query phrases. Thus, in making document vectors, we assign weights only to the elements which have non-zero value in the query phrase vector and bypass the remaining. To weight these elements, we use a weighting schema that very similar to *ltu*. The formula that we used for weighting the document vector elements is as follows:

$$Wd_{i,j} = \frac{\log(tf_{i,j}) * \log(\frac{N}{df_j})}{(1-slope)*pivot + slope * N.U.T}$$

Where $N.U.T$ is the number of unique terms of d_i, $Wd_{i,j}$ is weight of P_j in d_i. As can be seen, the *tf* and *idf* that we defined above differ from ordinary definitions of tf and idf in the sense that they are calculated for phrases instead of terms. In other words, document frequency of a phrase is the number of documents that contain all terms of the phrase. Term frequency of a phrase in a document is the number of occurrences of all terms of the phrase in the document. It can be considered as the minimum term frequency of phrase terms in the document. Therefore, document frequency and term frequency of a phrase intrinsically contain co-occurrences of phrase terms.

Our method works exactly like vector space model. We calculate the inner product of query phrase vectors with document vectors and rank the documents according to their similarity to query phrase vector.

3.2 Experiments

In some applications, such as Web search engines, effectiveness of information retrieval is tightly dependent on the precision of top ranked retrieved documents. For these applications, models with higher precision at top ranked documents and lower recalls are preferred. Thus, in our experiments we use precision at document cut-off measurement to compare the results.

Since we introduced our proposed model as an extension to the vector space model, we compare its performance with a vector space with ltn.ltu weighting schema. As BM25 has been shown to perform well in Persian text retrieval, we also compare it to our proposed model.

Our experiments for 50 queries of CLEF 2008, showed the higher precision of our proposed model at top ranked documents in comparison with both BM25 and Vector space. Similar results are observed for CLEF 2009. However, the improvement in precision for top ranked documents is more marked for the CLEF 2008 query set. Fig.1 shows the results of experiments for two sets of 50 queries.

It can be seen in Fig.1 that the proposed model has the highest precision compared to other models at lower recalls for CLEF 2008 query set. The precision of PBVS decreases for higher recalls but its MAP is comparable with the other IR models mentioned. In Fig.2 the precision of document retrieval for different document cut-offs is shown.

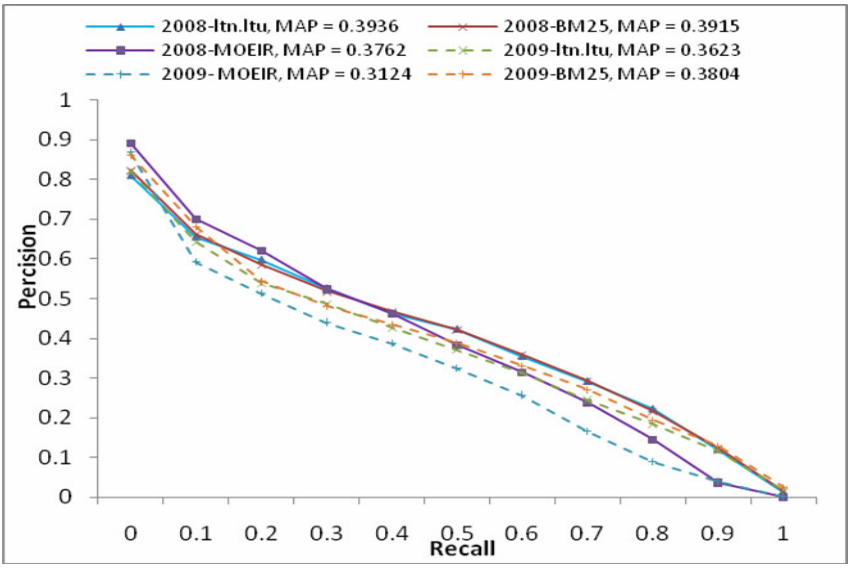


Fig. 1. Interpolated Precision-Recall for BM25, ltn.ltu and PBVS for 100 queries

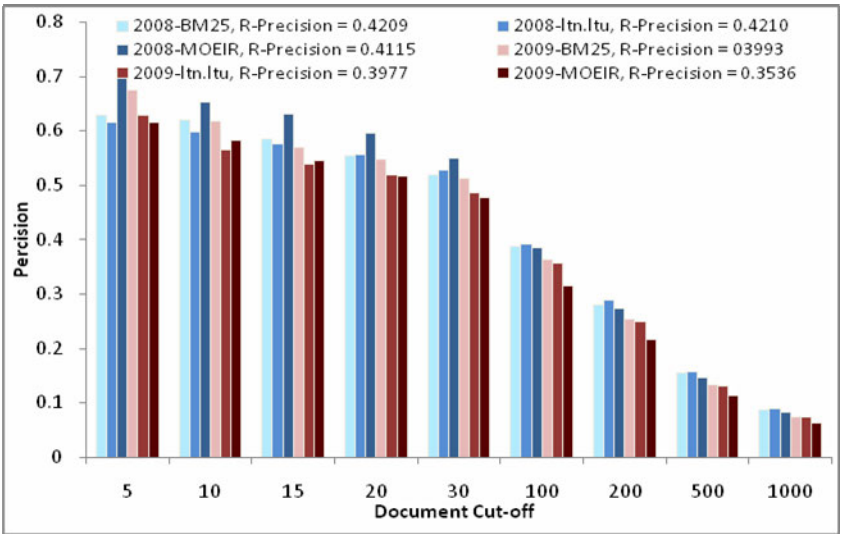


Fig. 2. Precision of different IR models at different document cut–offs for 100 queries

If we investigate the results of experiments query by query, we find that the performance of PBVS is not the best for all the queries. The experiments show that its performance is the best for 32 of 50 queries for CLEF 2008 and 21 of 50 queries for CLEF 2009. These 53 queries are those in which the main part of the information need is in the relations between terms and considering their terms separately leads to loss of a part of the information need represented by the query. As described, PBVS

maintains the semantic relations that exist within the query terms and thus its performance is better than the other models compared for these queries.

For example the 1st query of the CLEF 2009 query set is "حمله آمریکا به ایران – United States attack to Iran". This query contains 3 terms, حمله (Attack), آمریکا (United States) and ایران (Iran). All of these terms have a wide use in the Persian language. But when they are used together in the query they refer to a specific subject. So, in this query, the main information need is held in the query terms relations. Fig. 3 compares the performance of PBVS with a vector space model with ltn.ltu weighting schema for this query.

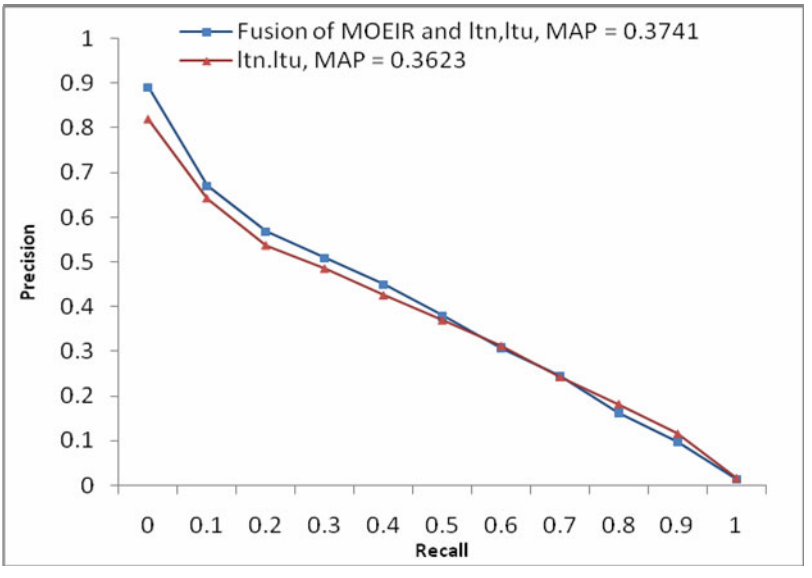


Fig. 3. Interpolated Precision-Recall for manually fused results of PBVS and vector space

Table 1. P@Cut-off Comparison of PBVS and ltn.ltu vector space for 3 queries of CLEF2009

Query	حمله آمریکا به ایران U.S attack to Iran		حقوق کودک Child's rights		خرید خدمت سربازي Military service selling	
Cut-off	ltn.ltu	PBVS	ltn.ltu	PBVS	ltn.ltu	PBVS
5	0.2000	0.6000	1.0000	1.0000	0.8000	1.0000
10	0.2000	0.6000	0.8000	1.0000	0.6000	0.9000
15	0.1333	0.6000	0.8000	0.9333	0.6667	0.9333
20	0.1500	0.5000	0.8000	0.9500	0.5000	0.8500
30	0.1333	0.4333	0.8667	0.9333	0.5000	0.6000
100	0.1700	0.3000	0.5900	0.6500	0.2300	0.2500

Table 1 shows the precision of retrieval at top retrieved documents for some significantly improved queries. For all of these queries some concatenation of query terms refers to a concept which differs from terms separately.

In Fig. 4, we fuse the results of the PBVS and vector space model manually. In this fusion, the results are drawn from PBVS for the queries which we already knew that it has better performance for them. For other queries results are drawn from the vector space model. As seen, the results have been improved for the CLEF09 query set.

Our experiments show that results of PBVS are better than other models for 53% of the queries. In these queries, a significant portion of the information need is represented in the relation of terms. Without consideration of these relations and ignoring the phrases, performance will be degraded. For other queries, for which PBVS's results were not good enough, consideration of query terms as phrases can bias the query term weightings in an incorrect manner and can thus decrease performance.

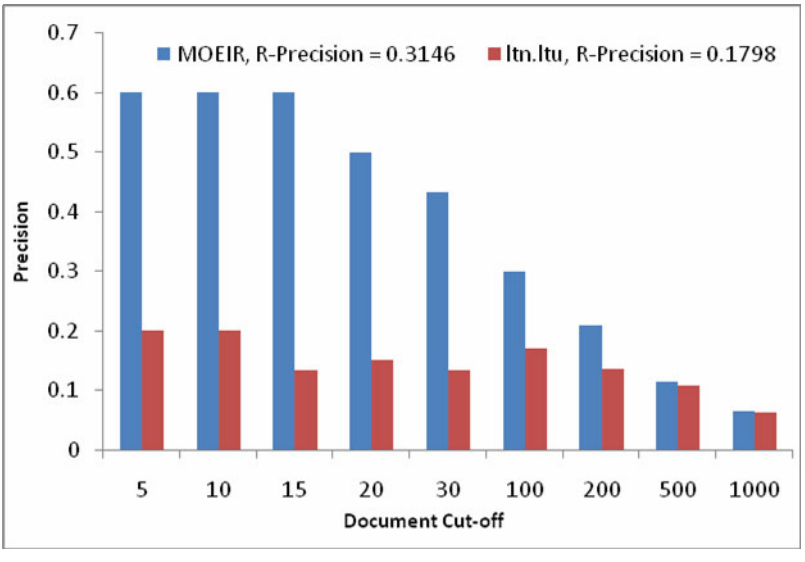


Fig. 4. Precision at different document cut-off for 1st query of CLEF09 query set

4 Conclusion

In this paper, we have presented an introduction to the Persian language and some resources for Persian IR studies. We have also proposed an IR model, which is a variation of the vector space model exploiting the relations existing between query terms. We have compared the performance of this model with some commonly used retrieval models, using CLEF 2008 and 2009 query sets. The results shows that the proposed model performs better at lower recall and document cut-offs. We observed that the CLEF 2009 query set is harder to process for IR models with respect to the CLEF 2008 query set.

Acknowledgments. The authors would like to thank the CLEF organization committee for their kind support. Also we would like to thank Iranian Telecommunication Research Center for supporting this research.

References

1. Gharib, A., Bahar, M., Fooroozanfar, B., Homaji, J., Yasami, R.: Farsi Grammar, 2nd edn., Jahane Danesh, Tehran (2001)
2. Taghva, K., Beckley, R., Sadeh, M.: A stemming algorithm for the Farsi Language. In: Proceedings of ITCC 2005, pp. 158–162. IEEE, Los Alamitos (2005)
3. Sharifloo, A., Shamsfard, M.: A Bottom up Approach to Persian Stemming. In: Proceedings of the Third International Joint Conference on NLP, Hyderabad, India (2008)
4. Mohammadi Nasiri, M., Sheikh Esmaeili, K., Abolhassani, H.: A statistical stemmer for the Persian language. In: Proceedings of 11th International Conference of Iranian Computer Society. Institute for Physics and Mathematics, Tehran (2005)
5. Tashakori, M., Meybodi, M., Oroumchian, F.: Bon: The Persian Stemmer. In: Shafazand, H., Tjoa, A.M. (eds.) EurAsia-ICT 2002. LNCS, vol. 2510, pp. 487–494. Springer, Heidelberg (2002)
6. AleAhmad, A., Amiri, H., Darrudi, E., Rahgozar, M., Oroumchian, F.: Hamshahri: A Standard Persian Text Collection. Journal of Knowledge-Based Systems 22(5), 382–387 (2009)
7. Riazati, D.: Computational Analysis of Persian Morphology. MSc thesis, Department of Computer Science, RMIT (1997)
8. Megerdoomian, K.: Unification-Based Persian Morphology. In: Proceedings of CICLing 2000, Alexander Gelbukh, Center of Investigation on Computation-IPN, Mexico (2000)
9. Megerdoomian, K.: Persian Computational Morphology: A unification-based approach, NMSU, CLR, Memoranda in Computer and Cognitive Science Report (2000)
10. Amiri, A., Hojjat, H., Oroumchian, F.: Investigation on a Feasible Corpus for Persian POS Tagging. In: Proceedings of 12th International CSI Computer Conference, Iran (2007)
11. Shamsfard, M., Hesabi, A., Fadaei, H., Mansoory, N., Famian, A., Bagherbeigi, S., Fekri, E., Monshizadeh, M., Assi, M.: Semi Automatic Development of FarsNet. In: The Persian WordNet, Global WordNet Conference, Mumbai, India (2010)
12. NLP lab at University of Tehran, `http://ece.ut.ac.ir/NLP/` (last visited on April 14, 2010)
13. Amtrup, J.W., Mansouri Rad, H., Megerdoomian, K., Zajac, R.: Persian-English Machine Translation: An Overview of the Shiraz Project. In: Memoranda in Computer and Cognitive Science, New Mexico State University (2000)
14. Dolamic, L., Fautsch, C., Savoy, J.: UniNE at CLEF 2008 TEL and Persian IR. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 178–185. Springer, Heidelberg (2009)

Combining Probabilistic and Translation-Based Models for Information Retrieval Based on Word Sense Annotations

Elisabeth Wolf[1], Delphine Bernhard[2,⋆], and Iryna Gurevych[1]

[1] Ubiquitous Knowledge Processing (UKP) Lab, Computer Science Department, Technische Universität Darmstadt, Germany
http://www.ukp.tu-darmstadt.de
[2] ILES group, LIMSI-CNRS, Orsay, France
delphine.bernhard@limsi.fr

Abstract. The objective of our experiments in the monolingual robust word sense disambiguation (WSD) track at CLEF 2009 is twofold. On the one hand, we intend to increase the precision of WSD by a heuristic-based combination of the annotations of the two WSD systems. For this, we provide an extrinsic evaluation on different levels of word sense accuracy. On the other hand, we aim at combining an often used probabilistic model, namely the Divergence From Randomness BM25 model, with a monolingual translation-based model. Our best performing system with and without utilizing word senses ranked 1st overall in the monolingual task. However, we could not observe any improvement by applying the sense annotations compared to the retrieval settings based on tokens or lemmas only.

1 Introduction

The CLEF robust WSD track 2009 follows the same design as in 2008, when runs by different systems were submitted varying in the pre-processing steps, indexing procedures, ranking functions, the application of query expansion methods, and the integration of word senses. In 2008, the best performance could be achieved by a combination of different probabilistic models (**PM**s) [7], namely the BM25 model, a Divergence From Randomness version of the BM25 model, and a statistical language model introduced by Hiemstra. In our experiments, we combine an often used PM with a monolingual translation-based model (**TM**), which was trained on definitions and glosses provided by different lexical semantic resources, namely WordNet, Wiktionary, Wikipedia, and Simple Wikipedia. This TM was successfully used for the task of answer finding by Bernhard and Gurevych [4]. Further, as all participants in 2008 took only one of the two systems for WSD into account when selecting the word sense annotations, we intend to increase the precision of WSD by an heuristic-based combination of the annotations of

⋆ This work was done while the author was at the UKP Lab, Darmstadt, Germany.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 120–127, 2010.

the two WSD systems. We provide an extrinsic evaluation on different levels of word sense accuracy. The task description and detailed information about the data collection can be found in the track overview paper [1].

2 Indexing and Retrieval Models

2.1 Indexing

We used Terrier (TERabyte RetrIEveR) [9], version 2.1 for indexing the documents. Each of the 169,000 documents is represented by its tokens. Each token is assigned a lemma and multiple word senses. Two different word sense disambiguation systems were applied, namely the UBC-ALM [2] and NUS-PT [5] system (abbreviated as UBC and NUS, respectively, in the remainder of the paper). In total, the document collection consists of approximately 100 million tokens including stop words. The NUS annotated corpus comes with around 199 million sense annotations including the sense probability scores, i.e. on average 2 senses per token. The UBC annotated corpus even consists of around 275 million sense annotations and probability scores, i.e. on average 2.75 senses per token. The accuracy of word sense annotations can highly influence the retrieval performance when utilizing word senses (see e.g. Sanderson [10]). Preliminary experiments on the training topics have shown that restricting the incorporated senses to the highest scored sense for each token increases the MAP of retrieval.

Further, we hypothesize that combining the NUS and UBC sense assignments increases the precision of annotated word senses. Therefore, we created several indices for our experiments. Each index consists of three fields, namely token, lemma, and sense. The indexed senses vary in the way they are selected. Four different indices were created: (i) an index with the highest scored UBC sense for each token (**UBCBest**), (ii) an index with the highest scored NUS sense for each token (**NUSBest**), (iii) an index with senses that were assigned by both systems and have the greatest sum of scores (**CombBest**), and finally (iv) an index with senses as in (iii), but where we chose the sense with the highest score from the UBC or NUS corpus when the intersection of the set of senses that were assigned by both systems is empty (**CombBest**$^+$). The construction of **CombBest** can be formally described by:

$$sense(t) = \underset{s \in S(t)}{\operatorname{argmax}} \quad score^{UBC}(s) + score^{NUS}(s) \tag{1}$$

with $S(t) = S^{UBC}(t) \cap S^{NUS}(t)$, where $S^{UBC}(t)$ is the set of senses of token t obtained from the UBC system and $S^{NUS}(t)$ is the sense set accordingly obtained from the NUS system. Thus, $S(t)$ is the intersection of the senses of token t annotated from the UBC and NUS systems. Further, $score^{UBC}(s)$ and $score^{NUS}(s)$ is the probability score assigned to sense s from the UBC and NUS system. If no probability score is assigned $score^{UBC}(s)$ and $score^{NUS}(s)$ returns 0, respectively. Accordingly, **CombBest**$^+$ is defined as:

$$sense(t) = \begin{cases} \operatorname{argmax}_{s \in S(t)} & score^{UBC}(s) + score^{NUS}(s) \quad \text{if } S(t) \neq \emptyset \\ \operatorname{argmax}_{s \in S^+(t)} & score^{UBC}(s) + score^{NUS}(s) \quad \text{otherwise} \end{cases} \tag{2}$$

where $S^+(t) = S^{UBC}(t) \cup S^{NUS}(t)$ is the union of the sense sets of token t from the UBC and NUS systems.

Prior to indexing, we applied standard stopword removal. Without stopwords, all indices consists of approximately 40.7 million tokens. As shown in the third column of Table 1 the UBCBest index contains around 34.1 million senses, the NUSBest index contains around 34.5 million senses, i.e. 6.6 million and 6.2 million tokens are not annotated with any sense in the UBCBest and NUSBest index, respectively. The CombBest index contains only 31.7 million senses, while the CombBest$^+$ index consists of 35.1 million senses.

2.2 Retrieval Models

We carried out several retrieval experiments using the Divergence From Randomness BM25 model (DFR_BM25). Often, such PMs have problems dealing with synonymy. This problem, also called *lexical gap*, arises from alternative ways of expressing a concept using different terms. Query expansion models try to overcome the lexical gap problem by reformulating the original query to increase the retrieval performance. We chose the the Kullback-Leibler (KL) query expansion model [6], since it performed best on the training data. In our experiments the original query is expanded by up to 10 most informative (highest weighted) terms from the 3 top ranked documents.

A further solution to the lexical gap problem is the integration of monolingual TMs first introduced by Berger and Lafferty [3]. These models encode statistical word associations which are trained on parallel monolingual document collections such as question-answer pairs. Recently, Bernhard and Gurevych [4] successfully applied TMs for the task of answer finding. In order to automatically train the TMs, they used the definitions and glosses provided for the same term by different lexical semantic resources, namely WordNet, Wiktionary, Wikipedia, and Simple Wikipedia yielding domain-independent TMs. The authors have shown that their models significantly perform better than baseline approaches for answer finding. In our experiments we employed the model defined by Xue et al. [11] and used by Bernhard and Gurevych [4]:

$$P(q|D) = \prod_{w \in q} P(w|d)\,, \tag{3}$$

where

$$P(w|d) = (1-\lambda)P_{mx}(w|d) + \lambda P(w|D)\,, \tag{4}$$

$$P_{mx}(w|d) = (1-\beta)P_{ml}(w|d) + \beta \sum_{t \in d} P(w|t)P_{ml}(t|d)\,, \tag{5}$$

q is the query, d the document, λ the smoothing parameter for the document collection D and $P(w|t)$ is the probability of translating a document term t to the query term w. The parameter β was set to 0.8 and λ to 0.5.

We applied the TM trained for the answer finding task, though it was not particularly trained for our task. As the TM was trained on tokens, we apply it on the indexed token field exclusively.

Table 1. Number of indexed word senses and MAP on retrieval (model: DFR_BM25 + KL) for different index types

index type	# senses	MAP
UBCBest	34.1 million	0.2636
NUSBest	34.5 million	0.3473
CombBest	31.7 million	0.3313
CombBest$^+$	35.1 million	**0.3551**

2.3 Combination of Retrieval Models

Our hypothesis is that TMs retrieve different documents for some queries than PMs. Therefore, we compute a combined relevance score to improve the retrieval performance. First, we normalize the scores resulting from each model applying standard normalization:

$$r_{norm}(i) = \frac{r_{orig}(i) - r_{min}}{r_{max} - r_{min}}, \tag{6}$$

where $r_{orig}(i)$ is the original score, r_{min} is the minimum, and r_{max} is the maximum occurring score for a query. Second, we combine the normalized relevance scores computed for individual models into a final score using the CombSUM method introduced by Fox and Shaw [8]. This method ranks the documents based on the sum of the (normalized) similarity scores of individual runs. Each run can be assigned a different weight.

3 Retrieval Results

3.1 Preliminary Experiments on Word Senses

As stated in Section 2.1 we created four indices which differ in the way word senses assigned by the UBC and NUS systems are selected. Table 1 shows the number of indexed word senses for the total number of 40.7 million tokens and the MAP values of different retrieval experiments applying the DFR_BM25 ranking model with KL query expansion. Retrieval on the UBCBest index shows a MAP value of 0.2636. For retrieval based on the NUSBest index the MAP value increases by 24.1%. According to this extrinsic evaluation, the NUS system clearly outperforms the UBC system. While CombBest does not increase the retrieval performance measured by MAP (0.3313), we were able to significantly[1] increase the MAP value using the CombBest$^+$ index up to 0.3551. In the remainder of this paper, we use the indices CombBest and CombBest$^+$ as our intention was to analyze the performance of the heuristic-based combination approach. The runs that we officially submitted are based on the CombBest index only.

[1] We used a two-tailed paired t-test ($\alpha < 0.05$) to determine the statistical significance.

Table 2. MAP values of the different retrieval models and index fields

retrieval model	token	lemma	sense CombBest	sense CombBest$^+$
TM	0.3616	-	-	-
DFR_BM25	0.3741	0.4054	0.2867	0.3096
DFR_BM25 + KL	0.4223	**0.4451**	0.3313	0.3551

3.2 Stand-Alone Retrieval Models

Table 2 shows the MAP of the different models. The TM is always restricted to the indexed tokens; the PM can use all different fields. We did not perform any fine-tuning on the parameters. The TM and the DFR_BM25 model without any query expansion show similar MAP values. However, when applying query expansion the DFR_BM25 approach outperforms the TM. The DFR_BM25 model with query expansion on tokens yields a MAP value of 0.4223 while we get a MAP value of 0.4451 on lemmas, which is an improvement of 5.1%. Experiments on senses achieve the lowest performance ranging from 0.2867 up to 0.3551. Applying query expansion on the CombBest and CombBest$^+$ index outperforms the runs without query expansion. In the following, we focus on experiments applying the DFR_BM25 model with query expansion (hereafter referred to as PM) and the TM.

3.3 Combination of Retrieval Models

We extensively experimented on the training data with different combination weights for the PM and TM using the CombSUM method described in Section 2.3. The combination achieves best performance when the PMs based on tokens and lemmas were assigned a higher weight (due to their higher MAP values) than the model based on senses or the TM. Table 3 illustrates the results obtained on the test topics by different combinations, with and without the integration of word senses. The presented weight combinations yielded best performance on the training data.

Two combinational aspects are of particular interest. The combination of the PMs based on tokens and lemmas yields no improvement (as no sense annotations are used CombBest and CombBest$^+$ yield equal performance). In contrast, the combinations of the PM with the TM always leads to an improvement. Even if the impact of the TM, i.e. its weight, is low (here: 0.2), the MAP values significantly increase when compared to the results obtained by the PM alone, on the token and lemma index fields. This fact corroborates our hypothesis that the PM and the TM retrieve different sets of relevant documents for some queries and that those different sets are effectively combined applying the CombSUM approach.

The second interesting aspect concerns the integration of word sense information. As listed in Table 1 retrieval based on senses from the CombBest index yields a MAP of 0.3313, while retrieval based on senses of the CombBest$^+$ index

Table 3. MAP values and weights for the combination of different models, using the CombBest and CombBest$^+$ indices. The settings marked with a '*' were submitted.

model:field	weights for combinations without word sense annotations				weights for combinations with word sense annotations				
TM:token	-	0.2	0.2	0.2	–	–	0.1	0.1	0.1
PM:token	0.5	0.8	-	0.4	0.8	–	0.8	–	0.4
PM:lemma	0.5	-	0.8	0.4	–	0.8	–	0.8	0.4
PM:sense	-	-	–	–	0.2	0.2	0.1	0.1	0.1
index type	MAP values				MAP values				
CombBest	0.4409	0.4316*	**0.4509***	0.4500*	0.4303	0.4461*	0.4330*	**0.4500***	0.4481*
CombBest$^+$					0.4327	0.4473	0.4331	**0.4507**	0.4480

shows a MAP of 0.3551. We attribute the difference to the fact that CombBest loses information about the documents due to the smaller amount of indexed senses. However, all combinations either with the CombBest or the CombBest$^+$ senses end up with a very similar performance. The reason could be that the loss of information when using the CombBest index is compensated by querying the tokens or lemmas as well.

In some combinational variations, the integration of word senses could achieve a higher MAP value than retrieval settings without word senses. For example, the MAP value corresponding to the retrieval based on tokens alone is 0.4223 (see Table 1), while the combination with senses obtains a MAP value of 0.4303 for the CombBest index and even 0.4327 for the CombBest$^+$ index. However, for the combination based on lemmas and senses, the difference is not significant. Overall, the best performance is obtained by the combination of the TM and the PM based on lemmas and senses, applying weights of 0.1, 0.8, and 0.1, respectively.

3.4 Discussion

In the previous section, we described all our experiments carried out on the document collection disambiguated with word senses. We submitted five runs without the integration of word senses and five further runs utilizing the annotated word senses. According to the MAP values our runs without word senses ended up in the 1st place out of 10 participants. Our highest MAP value could be achieved with the combination of the TM and the PM based on lemmas and the assigned weights of 0.2 and 0.8, respectively. When utilizing word senses, the combination of the TM and the PM based on both lemmas and senses obtains the 1st place according to the MAP as well. We mistakingly submitted runs on the CombBest index, even though we planned to focus on the CombBest$^+$ index. However, we have shown that the differences between the combinational approaches are minimal. Our best performing submitted retrieval setting achieved a MAP value of 0.4500, whereas the second top scoring system in the official challenge obtains a MAP value of 0.4346.

We increased the precision of WSD annotations through a heuristic-based combination of the UBC and NUS annotated senses, which we evaluated extrinsically. This evaluation has shown that the accuracy of annotated word senses highly influences the outcome of retrieval systems. However, we could not observe any improvement by applying the sense annotations compared to the retrieval settings based on tokens or lemmas only. This observation is consistent with the conclusion of last years' challenge.

Regarding the performance of the TM, the results on the combination are promising given that we merely applied a TM built for a previous application in the field of answer finding. The main drawback of the straightforward use is the discrepancy in the tokenization scheme. The tokenization of the document collection is not always compatible with the tokenization of the parallel corpora used for training the TM. In addition, the TM we used contains only tokens and thus cannot deal with indexed multiword expressions. For instance, the phrase "public transport" is indexed as "public_transport". In the TM the two terms "public" and "transport" appear, but not the phrase "public_transport". We quickly analyzed the amount of multiword expressions in the test topic collection. In fact, 61 queries out of the 160 test queries contain at least one multiword expression. This analysis shows that the TM was not particularly trained for this task and motivates further improvements. In addition, the TM could be trained on lemmas and senses. The latter option, however, requires a word sense disambiguated monolingual parallel corpus.

4 Conclusions

We have described a combinational approach to information retrieval on word sense disambiguated data, which combines a PM and a monolingual TM. For the PM we have used the DFR_BM25 model with the KL query expansion method. For the TM we have applied a model which was already trained for an answer finding task. Our aim was to assess the benefits of the combination of both models. We have shown that the combinational approach always achieves better performance than the stand-alone models.

Our second goal was to analyse different methods for selecting word senses from annotated corpora in order to increase their accuracy. We have discovered that our heuristic-based approach CombBest$^+$ increases the retrieval performance based on word senses by 2.2% when compared to NUSBest and even 25.8% when compared to UBCBest. The huge difference between NUSBest and UBCBest demonstrates that WSD accuracy is essential for utilizing word sense information. However, the experiments on the CombBest$^+$ index have shown that we could only increase the retrieval performance in one specific case: by combining the PM based on tokens with the same model based on senses. Nevertheless, other combinations without word senses outperformed this setting easily.

Acknowledgements

This work has been supported by the Emmy Noether Program of the German Research Foundation (DFG) under the grant No. GU 798/3-1, and by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under the grant No. I/82806.

References

1. Agirre, E., Di Nunzio, G.M., Mandl, T., Otegi, A.: CLEF 2009 Ad Hoc Track Overview: Robust-WSD Task. In: Multilingual Information Access Evaluation Text Retrieval Experiments. LNCS, vol. 1, Springer, Heidelberg (2010)
2. Agirre, E., Lopez de Lacalle, O.: UBC-ALM: Combining k-NN with SVD for WSD. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic, pp. 342–345 (2007)
3. Berger, A., Lafferty, J.: Information Retrieval as Statistical Translation. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 222–229 (1999)
4. Bernhard, D., Gurevych, I.: Combining Lexical Semantic Resources with Question & Answer Archives for Translation-Based Answer Finding. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore, pp. 728–736 (2009)
5. Chan, Y.S., Ng, H.T., Zhong, Z.: NUS-PT: Exploiting Parallel Texts for Word Sense Disambiguation in the English All-Words Tasks. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (Sem Eval-2007), Prague, Czech Republic, pp. 253–256 (2007)
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience, New York (1991)
7. Dolamic, L., Fautsch, C., Savoy, J.: UniNE at CLEF 2008: TEL, and Persian IR. In: Peters, C., et al. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 178–185. Springer, Heidelberg (2009)
8. Fox, E.A., Shaw, J.A.: Combination of Multiple Searches. In: Proceedings of the 2nd Text REtrieval Conference (TREC-2), pp. 243–252 (1994)
9. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Lioma, C.: Terrier: A High Performance and Scalable Information Retrieval Platform. In: Proceedings of the ACM SIGIR Workshop on Open Source Information Retrieval (2006)
10. Sanderson, M.: Word Sense Disambiguation and Information Retrieval. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, pp. 142–151 (1994)
11. Xue, X., Jeon, J., Croft, W.B.: Retrieval Models for Question and Answer Archives. In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, pp. 475–482 (2008)

Indexing with WordNet Synonyms May Improve Retrieval Results

Davide Buscaldi and Paolo Rosso

Natural Language Engineering Lab.,
ELiRF Research Group,
Dpto. de Sistemas Informáticos y Computación (DSIC),
Universidad Politécnica de Valencia, Spain
{dbuscaldi,prosso}@dsic.upv.es

Abstract. This paper describes a method developed for the Robust - Word Sense Disambiguation task at CLEF 2009. In our approach, a WordNet expanded index is generated from the disambiguated document collection. This index contains synonyms, hypernyms and holonyms of the disambiguated words contained in documents. Query words are integrated by terms extracted by means of a pseudo relevance feedback technique. The set of terms made of query words and terms resulting from pseudo relevance feedback are searched for in both the expanded WordNet index and the default index. The results show that the use of the extended index did not prove useful, obtaining $14-16\%$ less in MAP with respect to the base system. However, for some queries, expanding index terms with synonyms resulted particularly useful.

1 Introduction

The use of WordNet senses to improve the precision of Information Retrieval (IR) systems is probably one of the holy grails of modern research in IR. Since 1993, starting with the work of Ellen Voorhees [8], many researchers attempted to use effectively WordNet in IR, sometimes with good results [7], in other cases without success [3]. The Robust-WSD task introduced in CLEF 2009 represents an interesting attempt to foster further investigation in this field. In 2008, we participated in the QA-WSD task using an index expansion method based on WordNet hypernyms, synonyms and holonyms, which exploited the disambiguated collection [1]. The results did not show any relevant difference between the use of disambiguation or not, although we observed that passages returned using the disambiguated collection and our method tended to be shorter with respect to the base system. We took the opportunity presented by the Robust WSD Task at CLEF 2009 to test the same method in this task. A novelty for this participation was the introduction of a naïve Pseudo Relevance Feedback[5,9] method, consisting in the expansion of the query with the top 5 terms (according to their tf.idf weights) resulting from the unexpanded query.

In the following section, we describe the retrieval system. In section 3 we describe the characteristics of our submissions and discuss the obtained results.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 128–134, 2010.

2 Description of Retrieval System

The core of the system is a standard Lucene[1] search engine (version 2.4.1). During the indexing phase, we create two indices: the first one (*text*) containing all the terms of the sentence; the second one (expanded index, or *wn* index) containing all the synonyms of the disambiguated words (we consider the sense with the highest score to be the "right" sense). In the case of nouns and verbs, it contains also their hypernyms. For nouns, the holonyms (if available) are also added to the index, in a similar way to the GeoWorSE system that participated in the 2008 GeoCLEF track [2]. For instance, let us consider the following sentence from document `GH951115-000080`:

> Splitting the left from the Labour Party would weaken the battle for progressive policies inside the Labour Party.

The underlined words are those that have been disambiguated in the collection. For these words we can found their synonyms and related concepts in WordNet, as listed in Table 1.

Table 1. Expansion of the index terms of the example sentence. NA : not available (the relationship is not defined for the Part-Of-Speech of the related word).

lemma	ass. sense	synonyms	hypernyms	holonyms
split	4	separate part	move	NA
left	1	–	position place	–
Labour Party	2	labor party	political party party	–
weaken	1	–	change alter	NA
battle	1	conflict fight engagement	military action action	war warfare
progressive	2	reformist	NA	NA
policy	2	–	argumentation logical argument line of reasoning line	–

Therefore, the *wn* index will contain the following terms: *separate, part, move, position, place, labor party, political party, party, change, alter, conflict, fight, engagement, war, warfare, military action, action, reformist, argumentation, logical argument, line of reasoning, line*.

[1] `http://lucene.apache.org`

Previously to the search phase, in the default configuration, the original query is expanded with a naïve Blind Relevance Feedback (BRF) method. The *text* index is searched for question terms. The top 5 resulting documents are analysed to extract up to 5 keywords that are used to expand the original query. The keywords are selected according to their *tf.idf* weight (idf is calculated over the entire document collection). The expanded query is then submitted to the search engine to produce the final list of relevant documents.

In the WSD configuration, search is carried out in a similar way, with the difference that all nouns and adjectives are also searched for in the *wn* index.

In Table 2 we show the expansion terms obtained for topic `147-AH` : "*Oil accidents and birds*", using the two different configurations.

Table 2. Terms extracted for pseudo relevance feedback, topic 147-AH. Original query: "Oil accidents birds"

mode	term	tf.idf weight
No-WSD	gero	52.07
	pigeon	31.68
	fli	29.21
	spill	28.66
	wildlife	24.24
WSD	spill	200.60
	pipeline	174.10
	river	64.05
	arco	63.93
	fish	61.82

3 Experiments

We submitted four runs with the WSD system, two using the NUS labeled collection and two with the UBC labeled collection. For each collection, we submitted one run using only the topic title and another one using both the title and the description. As baseline, we submitted two non-WSD runs, one in the configuration "title only" and one in the configuration "title and description".

In Table 3 we show the results obtained by the two non-WSD runs and the four WSD runs.

The results show that the use of the disambiguated collection in general did not prove useful, since the base system obtained a better average MAP in all configurations. There are differences of $\sim 16\%$ in MAP between the normal and WSD runs in the title only configuration, and up to 14.21% between in TD configuration. The difference ($\sim 1\%$ in TD configuration) between the use of the NUS disambiguated collection and the UBC disambiguated collection is tiny, demonstrating that the disambiguation method is not relevant.

Table 3. Results obtained by our system at the CLEF 2009 Robust WSD track. TD: Title and Description. TO: Title Only. NUS: NUS labelled collection. UBC: UBC labelled collection. gm_AP: Geometric Mean Average Precision.

run ID	WSD	type	avg. MAP	avg. R-Prec	gm_AP
NLEL0901	n	TD	40.26%	38.72%	17.50%
NLEL0906	n	TO	33.42%	32.98%	8.75%
NLEL0902	y	TD NUS	27.14%	26.57%	6.87%
NLEL0904	y	TD UBC	26.05%	25.59%	6.42%
NLEL0903	y	TO NUS	17.48%	17.63%	1.14%
NLEL0905	y	TO UBC	17.53%	18.67%	1.24%

We analyzed some of the queries in which the standard system performed considerably better than the one which used the disambiguated collection. We find that disambiguation errors were the reason of the bad results. For instance, let us examine the results for topic `AH − 141`, "Letter Bomb for Kiesbauer": the base system obtained 100% MAP, placing the only relevant document (`GH950610 − 000164`) at the top of the list of retrieved documents, while the WordNet-based system obtained 0.4%, placing the relevant document only in the 255^{th} position. The best matching document, according to the WordNet-based system, was `LA010894 − 0146`, titled "Viacom, Blockbuster to merge in Paramount bid". The reason of this result is that all references to "*Blockbuster*" in the document were assigned to the first sense of "blockbuster" in WordNet: "a large bomb used to demolish extensive areas". The effect, given the indexing method, was to add the hypernym "general purpose bomb" to the *wn* index, and to obtain a high relevance for this document for any query containing "bomb", since "blockbuster" is very frequent in the document (and so its hypernym).

Although the overall results and the study of this cases demonstrate that, *in general*, expanding the index with WordNet is more harmful than useful, we observed that in 49 topics (30.6% of the total) the outcome of using WordNet for indexing was to obtain better MAP than in the case of not using it. In Figure 1 we show the difference in MAP for those topics.

We focused on the topics for which the MAP increment was more important - topic `AH − 180` "Bankruptcy of Barings" and topic `AH − 200` "Flooding in Holland and Germany". In the first case, in the WSD run, one of the extracted expansion terms was "Leeson", the surname of the person responsible for the bankruptcy. This term is very specific to the topic and was key to obtain the improvement for the WordNet-based system, since it appears in every relevant document. Looking for terms in the *wn* index allowed to find this expansion term that the base system could not find, mainly because of the term "bankruptcy", that did not appear in any of the relevant documents.

In topic `AH − 200`, *Holland* was not used in the relevant documents. One of its synonyms, *Netherlands*, was used instead. Since this term is listed in WordNet as one of the synonyms of "Holland", the WordNet-based system was able to found it in the expanded index.

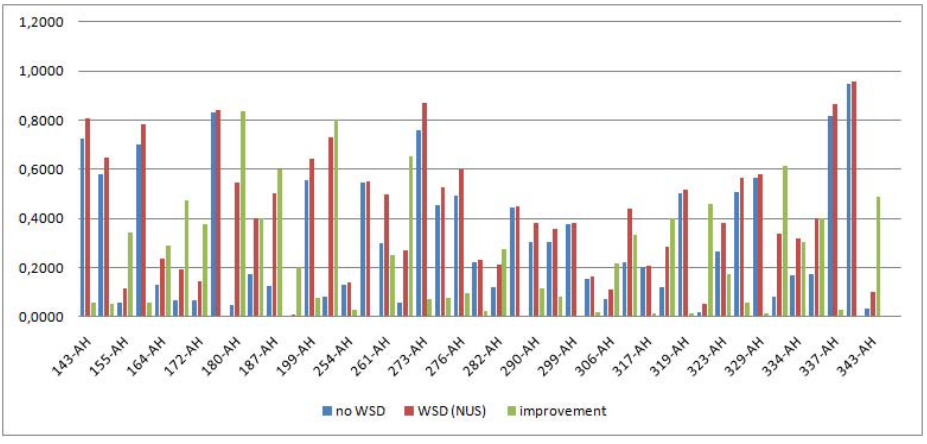


Fig. 1. Comparison of MAP values obtained for topics in which the use of the WordNet-based indexing method allowed to obtain better results than the base system

In order to understand which improvements were due to the relevance feedback method and which depended from the adopted WordNet-based expansion, we carried out some experiments with a system that did not include the BRF phase. The results of these experiments can be seen in Table 4.

Table 4. Results obtained by the system without blind relevance feedback

run ID	WSD	type	avg. MAP	avg. R-Prec	gm_AP
noBRF01	n	TD	30.56%	30.71%	10.72%
noBRF02	y	TD NUS	17.18%	18.15%	2.46%

These results shows a significant drop in precision with respect to the results obtained using BRF. In Figure 2 we show the precision increments obtained over the base system without relevance feedback (noBRF01 in Table 3) with the use of the WordNet expansion method (noBRF02 run) and the use of BRF without WordNet (this configuration corresponds to the NLEL0901 run).

As can be seen in Figure 2, in some topics the use of the WordNet-based method allowed to obtain a great MAP increase ($30 - 60\%$) over the base system with no contribution from BRF. We examined the results obtained in such topics in order to understand how they were produced. Topic `AH-183`, "Asian dinosaur remains" resulted in a $\sim 55\%$ increase over the baseline. The reason was that some relevant documents contained references to China and Mongolia, places where some dinosaur remains were found, but no reference to Asia. Therefore, the expansion method allowed to retrieve and rank better these documents because "Asia" was added to the wn index, as an holonym of China and Mongolia. The same reason is due to the increase obtained for topic `AH-266`,

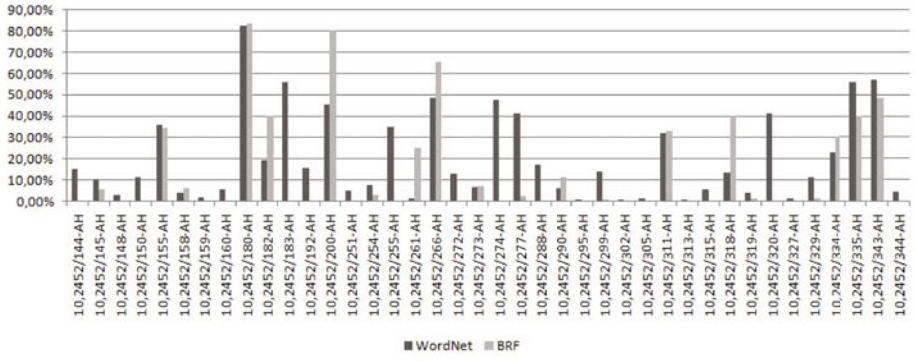


Fig. 2. MAP increases obtained with the WordNet-based indexing method and BRF, for topics in which WordNet resulted in an improvement over the base system MAP

"Discrimination against European Gypsies", where "Europe" was not included in the relevant documents, but only in their expansion. In topics `AH-255` ("Internet Junkies"), `AH-274` ("Unexploded World War II bombs") and `AH-277` ("Euthanasia by medics") the reason for the improvements was different, since the topic included both a term and one of its synonym (one in the title and the other in the description), but documents included only one of them. The synonyms were "addict" for "junky" in `AH-255`, "Second World War" for "World War II" in `AH-274` and "mercy killing" for "euthanasia".

4 Conclusions

The obtained results did not show any particular improvement depending on disambiguation accuracy. Sanderson's work [6] suggested that only high precision (more than 90%) in the disambiguation process may produce an improvement of the results in Information Retrieval. However, the collections used in the task were disambiguated with methods that, although being ones of the best systems available, were able to obtain $\sim 60\%$ in precision for the all-words task at Semeval 2007 [4]. Disambiguation errors proved to cause drops in retrieval accuracy, such in the case of topic $\texttt{AH} - \texttt{141}$, and errors were also propagated by the retrieval method selected: adding hypernyms and synonyms from errors introduced even more errors.

However, the analysis of the results obtained for some topics showed that a synonym of a non-ambiguous term was key to obtain an improvement, proving that at least in some cases, when terms are not ambiguous, adding synonyms to the index results useful. More research should be carried out on the cases in which disambiguation proved useful, in order to discover how the information extracted from WordNet affected the retrieval process. We should also check whether adding information only for not ambiguous terms could be better than expanding automatically disambiguated terms. Finally, we will need to determine

the cases in which the improvement was indirectly derived from terms extracted using the relevance feedback and those in which the use of WordNet related information affected directly the results.

Acknowledgements

We would like to thank the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 research project for partially supporting this work.

References

1. Buscaldi, D., Rosso, P.: Some experiments in question answering with a disambiguated document collection. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 442–447. Springer, Heidelberg (2009)
2. Buscaldi, D., Rosso, P.: Using geowordnet for geographical information retrieval. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 863–866. Springer, Heidelberg (2009)
3. Mihalcea, R., Moldovan, D.: Semantic indexing using wordnet senses. In: Proceedings of the ACL-2000 Workshop on Recent Advances in Natural Language Processing and Information retrieval, Morristown, NJ, USA, pp. 35–45. Association for Computational Linguistics (2000)
4. Pradhan, S.S., Loper, E., Dligach, D., Palmer, M.: Semeval-2007 task 17: English lexical sample, srl and all words. In: SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations, Morristown, NJ, USA, pp. 87–92. Association for Computational Linguistics (2007)
5. Robertson, S.E.: On term selection for query expansion. J. Doc. 46(4), 359–364 (1990)
6. Sanderson, M.: Word sense disambiguation and information retrieval. In: SIGIR 1994: Proceedings of the 17th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 142–151. Springer, New York (1994)
7. Schütze, H., Pedersen, J.O.: Information retrieval based on word senses. In: Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, pp. 161–175 (1995)
8. Voorhees, E.M.: Using wordnet to disambiguate word senses for text retrieval. In: SIGIR 1993: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 171–180. ACM, New York (1993)
9. Xu, J., Bruce Croft, W.: Query expansion using local and global document analysis. In: SIGIR 1996: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 4–11. ACM, New York (1996)

UFRGS@CLEF2009: Retrieval by Numbers

Thyago Bohrer Borges and Viviane P. Moreira

Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 - 91.501-970 - Porto Alegre - RS - Brazil
{tbborges,viviane}@inf.ufrgs.br

Abstract. For UFRGS's participation on CLEF's Robust task, our aim was to compare retrieval of plain documents to retrieval using information on word senses. The experimental runs which used word- sense disambiguation (WSD) consisted in indexing the synset codes of the senses which had scores higher than a predefined threshold. Several thresholds were tested. Our results have shown that the best WSD runs did not present a significant improvement in relation to the baseline run in which plain documents were used. In addition, a comparison between two alternative disambiguation systems has shown that one outperforms the other in all experimental runs.

1 Introduction

This paper reports on experiments submitted to CLEF 2009 Robust track. The aim of the task is to assess the validity of using word-sense disambiguated data for Information Retrieval (IR). Intuitively, the presence of ambiguity in IR is a cause for poor precision.

The interest in evaluating the aid of word-sense disambiguation (WSD) for IR is not recent. The first work [7] dates back to 1973 and was based on a set of rules. A comprehensive study by Krovetz & Croft [4] over the CACM and Time test collections found that resolving lexical ambiguity had little impact over retrieval effectiveness. Sanderson [5] experimented with the Reuters collection by synthetically adding ambiguity. He concluded that IR systems are more sensitive to wrong disambiguation than to ambiguity and that ambiguity only poses a problem for very short queries. Gonzalo et al. [3], on the other hand, found that indexing by WordNet synsets achieved a 29% improvement in relation to term indexing. This study used a corpus (SEMCOR) which was manually disambiguated. Furthermore, the authors assessed the impact of disambiguation errors and conclude that if the error rate surpasses 10%, retrieval effectiveness decreases. More recently, Stokoe et al. [6] carried out tests on the TREC W10G collection comparing term indexing to sense indexing and found a relative increase in precision of about 46%.

The goal of the experiments presented in this paper is to contribute data points to the evaluation of the benefits of WSD for IR. We also compare the two word-sense disambiguation systems used (UBC [1] and NUS [2]) in terms of their impact on retrieval effectiveness.

The remainder of this paper is organised as follows: Section 2 describes our experimental runs; Section 3 discusses our results and Section 4 summarises the conclusions.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 135–141, 2010.

2 Description of Runs and Resources

We worked on the English news collections composed by LA Times 94 and Glasgow Herald 95. Three versions of the collection were available: a "plain" version, and two versions with WSD data.

Using the WSD documents (UBC and NUS versions), we created document collections composed by the synset codes of all WordNet senses which exceeded an arbitrary threshold. WordNet is a lexical base, in which nouns, verbs, adjectives and adverbs are grouped in sets called "synsets". In our experiments, the threshold was systematically varied from 0.1 to 0.9 with increments of 0.1. As a result, we created 18 datasets - nine for NUS and nine for UBC.

Input	Output
<TERM ID="C041-27" LEMA="report" POS="VBP"> <WF>report</WF> <SYNSET SCORE="**0.393362015980332**" CODE="**00655029**-v"/> <SYNSET SCORE="0" CODE="00653609-v"/> <SYNSET SCORE="0" CODE="00653917-v"/> <SYNSET SCORE="0" CODE="00655324-v"/> <SYNSET SCORE="**0.606637984019668**" CODE="**00653371**-v"/> <SYNSET SCORE="0" CODE="00653772-v"/> </TERM>	00655029 00653371

Fig. 1. Original term with WSD information and the output of pre-processing

Figure 1 shows an example of an input word found in a document and the result of the pre-processing that extracts the synset codes with scores higher than 0.3. If a term did not have a synset code, or a sense scoring higher than the threshold, we kept the original word form (i.e. the contents of the <WF> tag). The same approach used in the documents was applied when building the queries from the topics.

The IR system we used was Zettair [8], which is a compact and fast search engine developed by RMIT University (Australia) distributed under a BSD-style license. Zettair implements a series of IR metrics for comparing queries and documents. We used Okapi BM25 as some preliminary tests performed on other data collections showed it achieved the best results. Our experiments did not employ stemming or stop-word removal since the document collections we generated contained mainly numbers. The time taken for indexing each data collection was approximately 1.5 minutes.

We produced a baseline run in which the plain collection is indexed and 18 runs using WSD-annotated documents. The WSD runs reported here are unofficial as they were prepared after the workshop. The details of the runs are shown in Table 1.

Over these data collections, 160 queries were performed, 153 of those had at least one relevant document in the collection. The time taken to run each set of 160 queries was approximately 20 seconds.

Table 1 shows that, as expected, by increasing the threshold for the synset score, the total number of terms decreases. The same happens with the average number of terms per document. As for the number of index terms, the behaviour is not monotonic, since we are keeping the original word forms for cases in which the term is not found in the WordNet or none of its senses surpasses the threshold.

Table 1. Details of the test collections

RunID	Number of distinct terms	Total number of terms	Average number of terms per document
Baseline	595,025	88,797,697	523
NUS-01	398,257	160,781,155	967
NUS-02	404,387	114,078,382	687
NUS-03	410,303	91,132,530	556
NUS-04	420,109	86,532,539	521
NUS-05	439,772	84,655,553	508
NUS-06	437,346	84,483,589	508
NUS-07	422,561	84,376,688	508
NUS-08	431,625	84,366,365	508
NUS-09	439,564	84,263,428	508
UBC-01	500,158	118,999,204	719
UBC-02	499,876	100,906,833	610
UBC-03	498,449	91,719,958	553
UBC-04	498,253	87,098,925	526
UBC-05	497,461	84,618,003	511
UBC-06	496,223	84,197,605	508
UBC-07	497,310	84,197,602	508
UBC-08	496,551	84,117,002	508
UBC-09	496,570	84,114,446	508

3 Results

Our results are summarised in Table 2. The results for the WSD run improve as the thresholds for synset scores increase. This happens because higher thresholds have the effect of keeping only the most probable sense of the words, working as a disambiguator. This is also an indicator that both NUS and UBC accurately assign synset scores.

Looking at absolute MAP figures, the baseline run outperformed nearly all WSD runs except from UBC-08 and UBC-09. UBC-09 was also slightly superior to the baseline in terms of GMAP. Considering Pr@10, the baseline run performed better than all WSD runs. The difference between the baseline and the best WSD runs is only marginal. This was confirmed by doing a T-test which showed no significant difference in terms of MAP, GMAP and Pr@10.

Table 2 also shows that UBC results are always better than their NUS counterparts. The difference between the pairs (e.g. UBC-01 vs. NUS-01) is statistically significant in all cases.

Figure 2 shows recall/precision curves for 6 of the 9 experimental runs (3 runs were omitted for space reasons). The curves clearly show the superiority of UBC in relation to NUS in this experimental setting. It is worth stressing that the aforementioned superiority was found in this particular experimental setting and that we have not performed a thorough intrinsic evaluation of the disambiguation systems that enable us to verify that one system is indeed better than the other. Our interest was to assess their contribution to improving retrieval effectiveness.

Table 2. Summary of Results

Run	MAP	Pr@10	GMAP
Baseline	0.3314	**0.3582**	0.1155
NUS-01	0.1271	0.1824	0.0157
NUS-02	0.1628	0.2248	0.0272
NUS-03	0.2008	0.2582	0.0412
NUS-04	0.2350	0.2889	0.0648
NUS-05	0.2488	0.2941	0.0677
NUS-06	0.2483	0.2922	0.0669
NUS-07	0.2549	0.2935	0.0688
NUS-08	0.2605	0.2974	0.0687
NUS-09	0.2619	0.2980	0.0687
UBC-01	0.2354	0.2725	0.0502
UBC-02	0.2747	0.2908	0.0716
UBC-03	0.3114	0.3553	0.1000
UBC-04	0.3244	0.3392	0.1082
UBC-05	0.3244	0.3399	0.1093
UBC-06	0.3236	0.3399	0.1071
UBC-07	0.3181	0.3131	0.1028
UBC-08	**0.3385**	0.3373	0.1140
UBC-09	0.3361	0.3477	**0.1163**

It is also possible to observe in Figure 2 that UBC runs start to approach the baseline as the threshold for the synset score is set to 0.4. When the score is set to 0.5, UBC runs outperform the baseline at low recall levels. However, this difference is not significant. Still it shows that WSD information enabled the IR to rank more relevant documents at the top.

A correlation test between the best WSD run and the baseline resulted in 0.85. This means that topics that do well in the plain run, also tend to do well in the WSD run.

We performed a topic-by-topic analysis considering the baseline run and the best WSD run (UBC-08) in terms of MAP. The analysis consisted in evaluating how many topics improved or worsened with WSD. If the difference in performance was less than 5% in MAP, we considered that there was a tie between the runs. The results of the analysis have shown that 49 topics improved with WSD information, 79 worsened with WSD and the remaining 30 had no substantial difference.

Table 3 shows the top ten topics which were helped by the addition of WSD information and Table 4 shows the ten topics that were most harmed.

We attribute the gain in performance to the accurate disambiguation of terms such as *Oscar* (topic 198), *bank* (topic 265), *ETA* (topic 306). More importantly, some multi-word terms such as *royal family* (topic 194), *human rights* (topic 185), *peace treaty* (topic 197), *Mexico City* (topic 327), and *World War II* (topic 274) were correctly identified and grouped into a single synset code. As a result, these multi- word terms were indexed as single units, while in the plain run, they were indexed as separate entries causing loss in semantics.

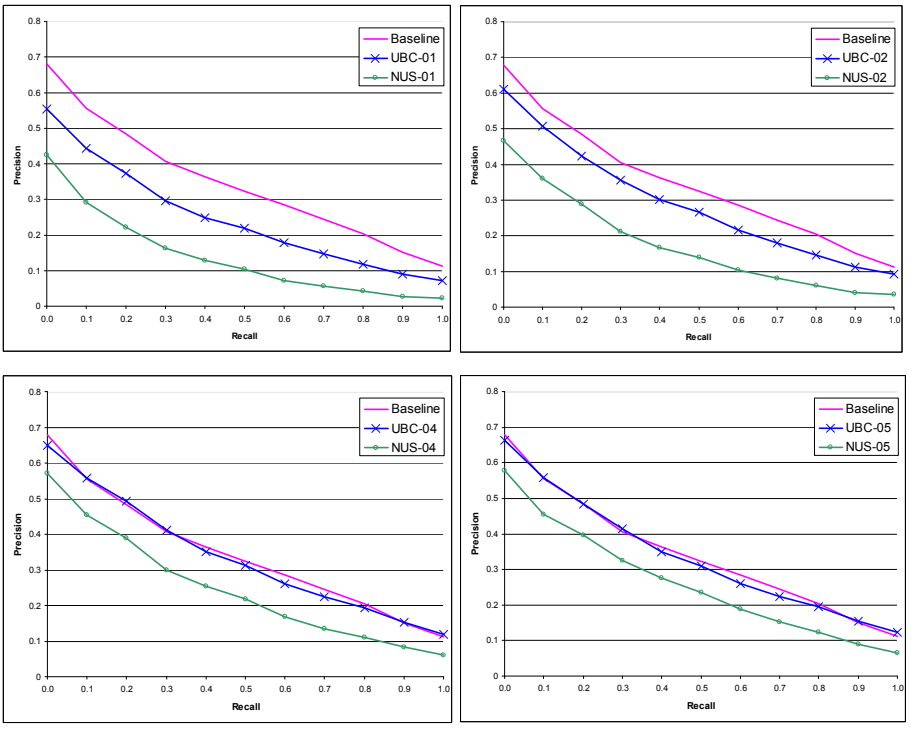


Fig. 2. Recall-Precision curves

Table 3. Ten topics with the biggest increase in MAP with the addition of WSD information

Topics	Baseline	UBC-08	Diff
10.2452/194-AH	0.1667	1.0000	0.8333
10.2452/198-AH	0.2500	1.0000	0.7500
10.2452/185-AH	0.3333	1.0000	0.6667
10.2452/265-AH	0.0954	0.6960	0.6006
10.2452/165-AH	0.5000	1.0000	0.5000
10.2452/306-AH	0.5000	1.0000	0.5000
10.2452/197-AH	0.4470	0.8298	0.3828
10.2452/182-AH	0.0447	0.3079	0.2632
10.2452/327-AH	0.0000	0.2135	0.2135
10.2452/274-AH	0.2055	0.4114	0.2059

As a general tendency, biggest improvement was attained by topics which had few relevant documents in the collection. These topics were helped by the correct disambiguation of polysemous terms and also by the correct treatment of multi-word terms which helped identifying the few relevant documents.

Table 4. Ten topics with the decrease increase in MAP with the addition of WSD information

Topics	Baseline	UBC-08	Diff
10.2452/264-AH	0.6152	0.3206	0.2946
10.2452/318-AH	0.3657	0.0859	0.2798
10.2452/340-AH	0.6393	0.3694	0.2699
10.2452/291-AH	0.4827	0.2356	0.2471
10.2452/175-AH	0.7473	0.5252	0.2221
10.2452/252-AH	0.2544	0.0454	0.209
10.2452/190-AH	0.3101	0.1085	0.2016
10.2452/349-AH	0.3872	0.1881	0.1991
10.2452/345-AH	0.2726	0.0762	0.1964
10.2452/350-AH	0.7287	0.5359	0.1928

We attribute the loss in performance in the WSD runs to the noise introduced by our choice of word senses. Because we only kept codes whose scores surpassed the threshold, and in some cases the correct sense was not the highest scoring, the procedure ended up causing erroneous disambiguation.

4 Conclusions

This paper described the experiments performed by our group for CLEF 2009 Ad hoc Robust task. We compared an experimental run in which we indexed the plain documents with 18 experimental runs in which we took WSD information into consideration. For our WSD experiments we indexed the synsets of words which exceeded a threshold which varied from 0.1 to 0.9. By using the synsets rather than the plain words, we were hoping to have a better representation of the contents of the documents and thus improve retrieval effectiveness.

The results of the experiments have shown that our best WSD runs marginally outperformed the baseline. This results are in line with previous research [4,5] which found no significant impact of WSD over retrieval effectiveness.

Our runs also compared the two disambiguation systems, NUS and UBC. We found out that, for our experimental setting, UBC is significantly better than NUS. However, it is worth pointing out that our aim was not to perform a thorough intrinsic evaluation of the disambiguation systems, instead we only assessed their impact on retrieval performance in a very particular experimental setting.

Acknowledgements

This work was partially supported by CNPq-Brazil.

References

1. Agirre, E., Lacalle, O.L.: UBC-ALM: Combining k-NN with SVD for WSD. In: SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations, Prague, pp. 342–345 (2007)

2. Chan, Y.S., Ng, H.T., Zhong, Z.: NUS-PT: exploiting parallel texts for word sense disambiguation in the English all-words tasks. In: Proceedings of the 4th International Workshop on Semantic Evaluations. Association for Computational Linguistics, Prague (2007)
3. Gonzalo, J., Verdejo, F., Chugur, I., Cigarrán, J.: Indexing with WordNet synsets can improve text retrieval. In: Proceedings of the COLING/ACL Workshop on usage of WorNet for NLP, Montreal, Canada, pp. 38–44 (1998)
4. Krovetz, R., Croft, W.B.: Lexical Ambiguity and Information Retrieval. ACM Transactions on Information Systems, TOIS (1992)
5. Sanderson, M.: Word Sense Disambiguation & Information Retrieval. In: Proceedings of the 17th International ACM SIGIR, Dublin, Ireland, pp. 49–57 (1994)
6. Stokoe, C., Oakes, M.P., Tait, J.: Word Sense Disambiguation in Information Retrieval Revisited. In: Proceedings of ACM SIGIR 2003, Toronto, Canada, pp. 159–166 (2003)
7. Weiss, S.F.: Learning to disambiguate. Information Storage and Retrieval 9, 33–41 (1973)
8. Zettair, (2007) www.seg.rmit.edu.au/zettair/ (retrieved 11/06/07)

Evaluation of Axiomatic Approaches to Crosslanguage Retrieval

Roman Kern[1], Andreas Juffinger[1], and Michael Granitzer[1,2]

[1] Know-Center, Graz
[2] Graz University of Technology
{rkern,ajuffinger,mgrani}@know-center.at

Abstract. Integrating word sense disambiguation into an information retrieval system could potentially improve its performance. This is the major motivation for the Robust WSD tasks of the Ad-Hoc Track of the CLEF 2009 campaign. For these tasks we have build a customizable and flexible retrieval system. The best performing configuration of this system is based on research in the area of axiomatic approaches to information retrieval. Further, our experiments show that configurations that incorporate word sense disambiguation (WSD) information into the retrieval process did outperform those without. For the monolingual task the performance difference is more pronounced than for the bilingual task. Finally, we are able to show that our query translation approach does work effectively, even if applied in the monolingual task.

1 Introduction

The intuition that determining the correct sense of ambiguous words could improve the performance of information retrieval systems has generated a lot of research in the last couple of years. Results in the area of monolingual retrieval could not life up to this expectations, see for example [1] and [2]. Short queries and the skewed distribution of senses partially explain for the observed results.

Despite moderate improvements for monolingual retrieval tasks, capabilities of systems using WSD information in other areas of information retrieval, like for example Question Answering (QA) and Cross-language Information Retrieval (CLIR), are still open questions. So for example in [3] the authors indicate that word sense disambiguation could help in multilingual retrieval.

For the CLEF2009 challenge we customized our retrieval system which has been developed for the CLEF2008 tasks, see [4]. This system has been modified to integrate different types of retrieval and ranking functions. The system contains multiple *TFIDF* weighting schemes, the BM25 [5] weighting function and additionally a retrieval function utilizing an axiomatic retrieval approach [6]. In our experiments we evaluated these different retrieval functions and different strategies to add the WSD information to the retrieval process.

Results show, that the best performing runs are based on the axiomatic retrieval approach. Further, runs incorporating WSD information did outperform those without, whereas for the monolingual task the performance difference is

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 142–149, 2010.

more pronounced than for the bilingual task. Finally, we are able to show that our query translation approach does work effectively, even if applied in the monolingual task.

The paper is structuered as follows: The next Section provides a detailed description of our system. In Section 3 their results of the various evaluation runs are presented and the main observations are discussed. Finally Section 4 concludes our findings.

2 Indexing and Retrieval System

Our information retrieval system consists of multiple separate components, firstly the CLEF article index, secondly the multilingual index, and thirdly, a query processing and document ranking unit.

2.1 CLEF Article Index

The document index is build using the collection of articles from the Los Angeles Times (1994) and the Glasgow Herald (1995) supplied by the organizers of the Robust WSD Task. These articles have already been tokenized and are annotated with senses using WordNet synsets. These senses are computed using two different word sense disambiguation systems - labeled UBC [7] and NUS [8]. We will report our results for both WSD sets separately in the evaluation section. For all terms that are annotated with multiple senses, we took the sense with the highest score. This sense, which is represented within WordNet as a synset, is added twice to the index. Once using its identifier and once using all synonyms within this synset. The query expansion using these features are labeled *Synset IDs* and *Synonyms* in the evaluation section. From the articles we only added the article body to the index. The headline of the articles were not processed as they did not appear to contribute to the relevance of the articles judging by results of the experiments made with our CLEF2008 system. No stop word removal was applied in the indexing stage.

Co-occurrence Term Statistics: By using WordNet and the annotated sense of ambiguous terms it is possible to determine the synonyms for a specific sense. The relation between synonymous words are one of many semantic relatedness relationship types between words. Statistical methods provide unsupervised means to detect word pairs with a high semantic relatedness without restriction to a specific relationship type. One of these methods is based on the co-occurrence statistics of words within a corpus. Many algorithms have been proposed to accomplish this task, using different weighting functions to measure the relationship between words. The Pointwise Mutual Information (PMI) has been found to provide good performance in this regard [9].

For our system, we implemented a query expansion technique based on the findings in [10]. Co-occurrence statistics based on the CLEF2009 article corpus were calculated by using a modified PMI measure for all words that occur at

least 2 times and less than in 50% of all documents. The similarity between two words w_i and w_j is defined as:

$$S_{CondPMI}(w_i, w_j) = \frac{log_2 \frac{P(w_i|w_j)}{P(w_j)}}{log_2(\frac{1}{P(w_j)})} \quad (1)$$

2.2 Multilingual Index

The multilingual index is used to translate individual terms from one language to another. This index can be created using various multilingual resources. We used two resources in our system, the *Wikipedia* and the *Europarl* corpus [11]. Both differ largely in their characteristics, such as domain and number of distinct terms. Another difference between the two resources is the alignment granularity. The Wikipedia multilingual index is aligned at the article level, whereas the Europarl corpus is aligned at the sentence level by applying the Church and Gale algorithm [12].

The goal of the multilingual index is to find the best matching terms in a language that is different to the original language of an input term using information retrieval techniques. The intuition behind our term translation approach is similar to select terms for query expansion using the top ranked documents in pseudo relevance feedback methods [13]. For each term, which can either be a single word or a phrase, a query is build. This query is then used to search for relevant documents in the query source language. From the top hits of the results - D_{top} - the aligned documents in the target language are retrieved. From the terms contained in these document the term candidates for translation are calculated.

We implemented three different scoring algorithms for estimating the best translation for the input term. The first is based on the well known *TFIDF* weighting scheme. For each term the weight w_i is calculated using the score of the most relevant documents D_{top} ($docFreq$ is the number of documents the translation candidate is contained in, N is the total number of documents):

$$w_i^{TFIDF} = log(\frac{N}{docFreq_i+1} + 1) * \sum_{d \in D_{top}} score(d) \quad (2)$$

The intuition behind the second scoring algorithm is to maximize the likelihood that a term has caused the document to be relevant. To accomplish this the same formula that is used to calculate the score of a document in the source language is applied on all target language terms found in the most relevant hits. The aggregated difference between the actual score and the reconstructed score serves as base for the weight of a single term:

$$w_i^{reconstruction} = \frac{1}{\sum_{d \in D_{top}} |tf_{i,d} * log(\frac{N}{docFreq_i+1}+1) - score(d)|+1} \quad (3)$$

The third scoring algorithm is based on the well-known cosine similarity. The vector of scores for the top scoring documents v^S in the result set is compared

with a vector v^i, which contains the $TFIDF$ weights calculated from the aligned document.

$$w_i^{cosine} = \frac{\sum_{d \in D_{top}} v_d^S v_d^i}{\|v^S\| \|v^i\|}, \text{where } v_d^i = tf_{i,d} * log(\frac{N}{docFreq_i + 1} + 1) \quad (4)$$

2.3 Query Processing and Document Ranking

The first step of the query processing is the selection of the topics parts used for query construction. In all our experiments we used the title and description part. The narrative section of the topics was not included in the query generation process. All terms were stemmed using the Snowball stemmer.

If the language of the topic differs from the language of the articles, the query terms are individually translated. The top n candidates according to the weighting function were then added to the query as translation for a single query term. Based on the training topics and relevance judgments we found that using only the two highest scoring translation terms to offer the best overall performance.

The result of the query generation is an unordered list of terms. In the next step relevant documents are retrieved and ranked. The *TFIDF* [14] weighting scheme and the *BM25* [5] approach are textbook methods to this problem and demonstrated robust and reliable performance in the past. A variant of the *TFIDF* retrieval model did provide good, but not state-of-the-art performance in the CLEF2008 Robust WSD task [4]. Many of the CLEF2008 participants incorporated the BM25 approach into their retrieval systems with great success (for example [15]). We therefore also report the performance of our system using an implementation of the BM25 weighting scheme[1]. The two weights k_1 and b were estimated by experiments on the training topics.

For our main experiments we have chosen to apply findings in the area of axiomatic approaches to information retrieval. Fang and Zhai present in [6] several variations of weighting functions build using a set of axioms that constrain the properties of a weighting function. The authors did recommend one of their derived retrieval functions which has shown promising performance in their evaluation. We did adapt this function for our retrieval system. The score of a document D out of N documents given a set of query terms Q is build using the tuning parameter α and β:

$$S_{Axiomatic}(Q, D) = \sum_{t \in Q \cap D} (\frac{N}{docFreq_t})^{\alpha} \times \frac{tf_{t,D}}{tf_{t,D} + 0.5 + \beta \frac{docLength_D}{avgDocLength}} \quad (5)$$

Using the training topics we found the setting of 0.25 for α and 0.75 for β to provide a satisfying performance.

3 Results and Discussion

The main motivation for the Robust WSD task is to measure the performance impact of using word sense disambiguation as part of a information retrieval

[1] http://nlp.uned.es/~jperezi/Lucene-BM25/

system. A first step to determine the influence of WSD information is the creation of a state-of-the-art retrieval system that does not incorporate a disambiguation process. We tried to build such a system and then use the WSD information as an optional processing step using query expansion. The results of these two system configurations should provide insights into the influence of word sense disambiguation. To further increase the validity of the observed behavior we also report the performance of our system using query expansion based on co-occurrence term statistics.

3.1 Monolingual Performance

In Table 1 different retrieval functions are compared using the CLEF2009 test collection without query expansion. Although this comparison gives no insights into the question whether WSD information could improve the performance, it demonstrates that the results of the axiomatic approach is indeed a valuable contribution to the arsenal of information retrieval techniques. The according GMAP measure is improved over the BM25 run, which indicates that especially low performing topics did improve using the axiomatic approach.

For the comparison with the configurations that utilize the WSD information we only report the performance figures achieved using the axiomatic retrieval function. Table 2 lists the performance measures of the various query expansion configurations. The best performing configuration combines the synonym, synset and term co-occurrence information, labeled "all" in the table. The performance figures do show that integrating the words sense disambiguation data into the retrieval process of our system does improve performance. Not only does the baseline configuration benefit from the sense annotations, but also the configuration that already uses a (successful) query expansion technique is improved further. The difference between the two WSD data sets (NUS and UBC) and between the Synonym and the Synset features are too small to allow any conclusions. We conducted two significance test to assess whether the improvement over the baseline (the axiomatic configuration for the first five runs and the query expansion using co-occurrence statistics for the last two runs) is statistically significant. These tests were the Wilcoxon signed rank test and randomized tests, which according to [16] should be the preferred way to test for significance in information retrieval applications.

Table 1. Performance of the monolingual system without query expansion

Retrieval Function	MAP	GMAP	Notes
TFIDF1	0.3083	0.1182	*Lucene Boolean Query*
TFIDF2	0.3313	0.1331	*Lucene Disjunction Max Query*
BM25	0.3889	0.1566	*Using* $k_1 = 0.8$ *and* $b = 0.5$
Axiomatic	0.4022	0.1805	*Using* $\alpha = 0.25$ *and* $\beta = 0.75$

Table 2. MAP and GMAP measures of the monolingual system using a combination of features together with the p-values of two significance test. Both significance tests agree that for most of the configurations the improvement is not achieved by chance.

Query Expansion	MAP	GMAP	Wilcoxon	Randomized
Synonyms (NUS)	0.4061	0.1849	0.0376	0.0517
Synonyms (UBC)	0.4036	0.1837	0.3286	0.2070
Synset IDs (NUS)	0.4047	0.1856	0.0303	0.0944
Synset IDs (UBC)	0.4070	0.1869	0.0119	0.0147
Co-occurrence Terms	0.4170	0.1864	0.0001	0.0196
All (NUS)	0.4222	0.1947	0.0174	0.0554
All (UBC)	0.4212	0.1942	0.1603	0.0730

3.2 Bilingual Performance

For the Spanish topics of the Robust WSD task we added the translation step into the query processing as described in Section 2.2. This processing step is executed prior to the query expansion step. When adding the WSD information to the retrieval process the performance of the system is increased, see Table 3. The gap between the best configuration and the baseline is just about 1%. The difference between the configuration that incorporate the WSD information are not statistically significant better than their respective baseline according to the two significance tests.

Translation Impact: For our final evaluation runs we investigated the impact of the query translation step. The motivation for this are the findings in [4] where the authors state that the query translation did not cause serious performance deterioration even if both the query and the documents are of the same language. Figure 1 summarizes the performance of our system using different languages and query translation functions. The results demonstrate that using our approach to translate a query does not have a pronounced negative effect on the retrieval performance when using the best performing translation strategy.

Table 3. Performance of the bilingual system using a combination of features. The p-values of the significance tests are lower for the bilingual results, with the exception of the co-occurrence query expansion.

Query Expansion	MAP	GMAP	Wilcoxon	Randomized
No Query Expansion	0.2885	0.0762		
Synonyms (1st)	0.2923	0.0762	0.0910	0.0888
Synset IDs (1st)	0.2933	0.0773	0.2187	0.0056
Co-occurrence Terms	0.2917	0.0718	0.0090	0.0252
All (1st)	0.2982	0.0746	0.2859	0.0611

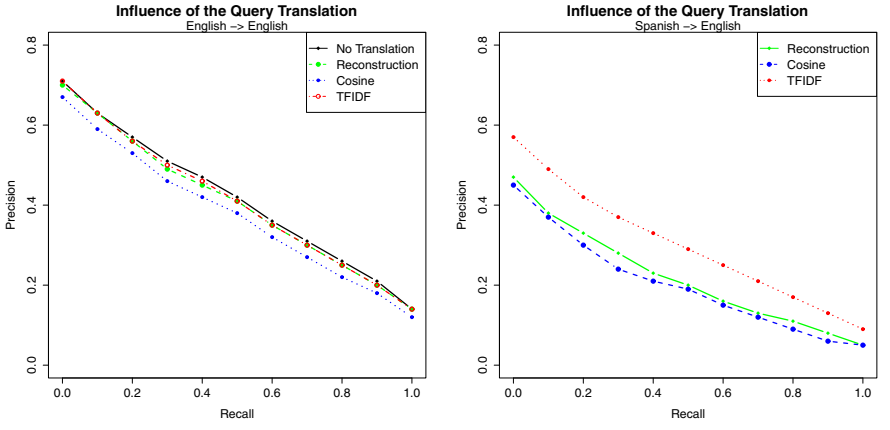


Fig. 1. Performance of the different translation functions for the English queries (left) and the Spanish queries (right). The optional translation step for the English queries has no noticable impact on the performance, at least when using the *TFIDF* or *Reconstruction* scoring method. For Spanish queries, which require a translation step, the *TFIDF* scoring algorithm outperforms the other two approaches by a margin.

4 Conclusion

In order to investigate the influence of words sense disambiguation in the area of cross language retrieval we built a system that can be operated in a number of configurations. This system was designed in a way to also study the performance of different retrieval functions. Additionally to the well known *TFIDF* weighting scheme and the *BM25* ranking function we adapted a retrieval function that has been developed using an axiomatic approach to information retrieval. This method did provide the best performance in our tests. For the bilingual retrieval task we developed a translation mechanism based on the freely available Wikipedia and the Europarl corpus.

In our evaluation runs we have found that incorporating the word sense disambiguation information does indeed improve the performance of our system by a small margin. This was the case for the monolingual and the bilingual task, although for the bilingual task the improvements are not statistically significant. Also when using the WSD information additionally to an existing query expansion technique the performance was further improved. In none of our tests we observed that the performance did decrease when applying the word sense disambiguation information. Just on a few queries there has been a negative impact. The reason for this and possible means to detect and to avoid poor performing queries are still open questions and require further research.

Acknowledgements

The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Sanderson, M.: Word sense disambiguation and information retrieval. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 142–151. Springer, New York (1994)
2. Voorhees, E.: Natural language processing and information retrieval. In: Pazienza, M.T. (ed.) SCIE 1999. LNCS (LNAI), vol. 1714, pp. 32–48. Springer, Heidelberg (1999)
3. Oard, D., Dorr, B.: A survey of multilingual text retrieval (1998)
4. Juffinger, A., Kern, R., Granitzer, M.: Exploiting Cooccurrence on Corpus and Document Level for Fair Crosslanguage Retrieval (2008)
5. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-4. In: Proceedings of the Fourth Text Retrieval Conference, pp. 73–97 (1996)
6. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 480–487. ACM, New York (2005)
7. Agirre, E., de Lacalle, O.: UBC-ALM: Combining k-nn with SVD for WSD. In: Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval, Czech Republic, Prague (2007)
8. Chan, Y., Ng, H., Zhong, Z.: NUS-PT: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In: Proceedings of SemEval, pp. 253–256 (2007)
9. Turney, P.: Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. LNCS (LNAI), pp. 491–502. Springer, Heidelberg (2001)
10. Terra, E., Clarke, C.: Scoring missing terms in information retrieval tasks. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, pp. 50–58. ACM, New York (2004)
11. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT summit, vol. 5 (2005)
12. Gale, W., Church, K.: A program for aligning sentences in bilingual corpora. Computational linguistics 19(1), 75–102 (1994)
13. Manning, C., Raghavan, P., Schtze, H.: Introduction to information retrieval. Cambridge University Press, New York (2008)
14. Salton, G., Buckley, C.: Term weighting approaches in automatic text retrieval (1987)
15. Dolamic, L., Fautsch, C., Savoy, J.: UniNE at CLEF 2008: TEL, Persian and Robust IR. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 178–185. Springer, Heidelberg (2009)
16. Smucker, M., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 623–632. ACM, New York (2007)

UNIBA-SENSE @ CLEF 2009: Robust WSD Task

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro

Department of Computer Science
University of Bari
70126 Bari, Italy
{basilepp,acaputo,semeraro}@di.uniba.it

Abstract. This paper presents the participation of the semantic N-levels search engine SENSE at the CLEF 2009 Ad Hoc Robust-WSD Task. Our aim is to demonstrate that the combination of the N-levels model and WSD can improve the retrieval performance even when an effective retrieval model is adopted. To reach this aim, we worked on two different strategies. On one hand a model, based on Okapi BM25, was adopted at each level. On the other hand, we integrated a local relevance feedback technique, called Local Context Analysis, in both indexing levels of the system (keyword and word meaning). The hypothesis that Local Context Analysis can be effective even when it works on word meanings coming from a WSD algorithm is supported by experimental results. In monolingual task MAP increased of about 2% exploiting disambiguation, while GMAP increased from 4% to 9% when we used WSD in both mono- and bi- lingual tasks.

1 Introduction

In this paper we present our participation at the CLEF 2009 Ad Hoc Robust-WSD Task. Our retrieval system is based on SENSE [2], a semantic search engine which implements the N-levels model. For the CLEF 2009 experiments, the following levels were exploited:

Keyword level - the entry level in which a document is represented by the words occurring in the text.

Word meaning level - at this level a document is represented through *synsets* obtained by WordNet, a semantic lexicon for the English language. A synset is a set of synonym words (with the same meaning).

SENSE is able to manage different models for each level. In CLEF 2008 campaign we adopted the standard Vector Space Model implemented in Lucene for both the keyword and the word meaning level. For CLEF 2009 our goal is to improve the overall retrieval performance by adopting a more powerful model, called Okapi BM25, and a pseudo-relevance feedback mechanism based on Local Context Analysis.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 150–157, 2010.

The rest of the paper is structured as follows: the indexing step adopted in SENSE is described in Section 2, while Section 3 presents the searching step with details about Local Context Analysis strategy. The details of the system setup for the CLEF competition are provided in Section 4. Finally, the experiments are described in Section 5. Conclusions and future work close the paper.

2 Indexing

In CLEF Ad-Hoc WSD Robust track, documents and queries are provided in XML format. In order to index the documents and read the queries we developed an XML parser using the XMLBeans[1] tool. As SENSE supports an indefinite numbers of levels, we developed a flexible indexing mechanism. Hence, we produced an intermediate data format which contains all the data necessary to the N-levels model. For each token this format provides a set of features needful to build each level. In CLEF, for the keyword level the stemming of the word[2] is provided, for the meaning one we provided the list of all possible meanings with the corresponding score. During the indexing we performed several text operations. One is stop words elimination. We built two different stop words lists, one for documents and one for queries. In this way we removed irrelevant words from queries. Moreover, before storing each token in a document, we replaced all occurrences of not alphanumeric characters with a single underscore character "_". This text normalization operation was also performed for queries during the search process. With respect to the meaning level, we index for each token only the WordNet synset with the highest score. For each document a bag of synsets is built. Consequently, the vocabulary at this level is the set of distinct synsets recognized in the collection by the WSD procedure.

3 Searching

The local similarity functions for both the meaning and the keyword levels are computed using a modified version of the Lucene default document score, that implements the Okapi BM25 [7]. In order to implement BM25 in SENSE we exploited the technique described in [5]. In particular, we adopted the BM25-based strategy which takes into account multi-field documents. Indeed, in our collection each document is represented by two fields: HEADLINE and TEXT. The multi-field representation reflects the XML structure of documents provided by the organizers. Table 1 shows the BM25 parameters used in SENSE, where avl is the average length for each field. b is a constant related to the field length, similar to b constant in classical BM25 formula, k_1 is a free parameter, while $boost$ is the boost factor applied to that field. All parameters were tuned on the training data and are different for keyword and meaning level.

[1] http://xmlbeans.apache.org/
[2] Stemming is performed by the Snowball library.

Table 1. BM25 parameters used in SENSE

Level	Field	k_1	N	avl	b	$boost$
Keyword	$HEADLINE$	3.25	166,726	7.96	0.70	2.00
	$TEXT$	3.25	166,726	295.05	0.70	1.00
Word Meaning	$HEADLINE$	3.50	166,726	5.94	0.70	2.00
	$TEXT$	3.50	166,726	230.54	0.70	1.00

For the meaning level, both query and document vectors contain synsets instead of keywords.

In SENSE each level produces a list of documents ranked according to the similarity function defined for that level (*local similarity function*). Since the ultimate goal is to obtain a *single* list of documents ranked in decreasing order of relevance, a *global ranking function* is needed to merge all the result lists that come from each level. This function is independent of both the number of levels and the specific local scoring and similarity functions because it takes as input n ranked lists of documents and produces a unique merged list of the most relevant documents.

The aggregation of lists in a single one requires two steps: the first one produces the n normalized lists and the second one merges the n lists in a single one. The two steps are thoroughly described in [2]. In CLEF we adopt Z-Score normalization and CombSUM [3,4] as score normalization and rank aggregation function, respectively. Each level can be combined using a different weighting factor in order to give different relevance to each level.

3.1 Query Expansion and Term Reweighting

We extended the SENSE architecture by integrating a query expansion module, as well as a technique for term reweighting. We adopted the Local Context Analysis (LCA) [8], a strategy that proved its effectiveness on several test collections. LCA is a *local* technique as it analyzes only the first top-ranked documents that are assumed to be the relevant ones. LCA relies on the hypothesis that terms frequently occurring in the top-ranked documents frequently co-occur with all query terms in those documents too. We employed the LCA for both levels exploited in our experiments: keyword and word meaning. The underlying idea is that the LCA hypothesis could also be applied to the word meaning level, in which meanings are involved instead of terms. Therefore, we extended the original measure of co-occurrence degree in order to weigh a generic feature (keyword or word meaning) rather than just a term. According to the original formula, we define the following function:

$$codegree(f, q_i) = \frac{log_{10}(co(f, q_i) + 1) * idf(f)}{log_{10}(n)} \quad (1)$$

codegree measures the degree of co-occurrence between the feature f and the query feature q_i ($co(f, q_i)$), but it takes also into account the frequency of f in

the whole collection ($idf(f)$) and normalizes this value with respect to n, the number of documents in the top-ranked set.

$$co(f, q_i) = \sum_{d \in S} tf(f, d) * tf(q_i, d) \tag{2}$$

$$idf(f) = min(1.0, \frac{log_{10} \frac{N}{N_f}}{5.0}) \tag{3}$$

where $tf(f, d)$ and $tf(q_i, d)$ are the frequency in d of f and q_i respectively, S is the set of top-ranked documents, N is the number of documents in the collection and N_f is the number of documents containing the feature f. For each level, we retrieve the n top-ranked documents for a query q by computing a function lca for each feature in the results set, as follows:

$$lca(f, q) = \prod_{q_i \in q} (\delta + codegree(f, q_i))^{idf(q_i)} \tag{4}$$

This formula is used to rank the list of features that occur in the top-ranked documents; δ is a smoothing factor and the power is used to raise the impact of rare features. A new query q' is created by adding the k top ranked features to the original query, where each feature is weighed using the lca value. Hence, the new query is re-executed to obtain the final list of ranked documents for each level. Differently from the original work, we applied LCA to the top ranked documents rather than passages[3]. Moreover, no tuning is performed over the collection to set the parameters. For the CLEF experiments, we decided to get the first ten top-ranked documents and to expand the query using the first ten ranked features. Finally, we set up the smoothing factor to 0.1 in order to boost those concepts that co-occur with the highest number of query features.

4 System Setup

We exploited the SENSE framework to build our IR system for the CLEF evaluation. We used two different levels: keyword (using word stems) and word meaning (using WordNet synsets). All SENSE components involved in the experiments are implemented in Java using the version 2.3.2 of Lucene API. Experiments were run on an Intel Core 2 Quad processor at 2.6 GHz, operating in 64 bit mode, running Linux (UBUNTU 9.04), with 4 GB of main memory.

Following CLEF guidelines, we performed two different tracks of experiments: Ad Hoc Robust-WSD monolingual and bilingual. Each track required two different evaluations: with and without synsets. We exploited several combinations between levels and the query relevance feedback method, especially for the meaning level. All query building methods are automatic and do not require manual operations. Moreover, we used different boosting factors for each topic field and gave more importance to the terms in the fields TITLE and DESCRIPTION.

[3] In the original work, passages are parts of document text of about 300 words.

Table 2 shows all performed runs. In particular, the column *N-Levels* reports the different weighting factors used to merge each result list. The columns *WSD* and *LCA* denote in which runs the word meaning level and pseudo-relevance feedback technique were involved. Details about boosting factors assigned to each query field are reported in the *Boost* column (T=Title, D=Description, N=Narrative). More details on the track are reported in the track overview paper [1]. For all the runs we removed the stop words from both the index and the topics.

Table 2. Overview of experiments

RUN	**Mono**	**Bi**	**N-levels**		**WSD**	**LCA**	**Boost**		
			Key	Syn			T	D	N
unibaKTD	X	-	-	-	-	-	8	1	-
unibaKTDN	X	-	-	-	-	-	8	2	1
unibaKRF	X	-	-	-	-	X	8	2	1
unibaWsdTD	X	-	-	-	X	-	8	1	-
unibaWsdTDN	X	-	-	-	X	-	8	2	1
unibaWsdNL0802	X	-	0.8	0.2	X	-	8	2	1
unibaWsdNL0901	X	-	0.9	0.1	X	-	8	2	1
unibaKeySynRF	X	-	0.8	0.2	X	X	8	2	1
unibaCrossTD	-	X	-	-	-	-	8	1	-
unibaCrossTDN	-	X	-	-	-	-	8	2	1
unibaCrossKeyRF	-	X	-	-	-	X	8	2	1
unibaCrossWsdTD	-	X	-	-	X	-	8	1	-
unibaCrossWsdTDN	-	X	-	-	X	-	8	2	1
unibaCrossWsdNL0802	-	X	0.8	0.2	X	-	8	2	1
unibaCrossWsdNL0901	-	X	0.9	0.1	X	-	8	2	1
unibaCrossKeySynRF	-	X	0.8	0.2	X	X	8	2	1

5 Experimental Session

The experiments were carried out on the CLEF Ad Hoc WSD-Robust dataset derived from the English CLEF data, which comprises corpora from "Los Angeles Times" and "Glasgow Herald", amounting to 166,726 documents and 160 topics in English and Spanish. The relevance judgments were taken from CLEF. Our evaluation has two main goals:

1. to prove that the combination of two levels outperforms a single level. Specifically, we want to investigate whether the combination of keyword and meaning levels turns out to be more effective than the keyword level alone, and how the performance varies.
2. to prove that Local Context Analysis improves the system performance. We exploit pseudo-relevance feedback techniques in both levels, keyword and meaning. Our aim is to demonstrate the effectiveness of pseudo-relevance feedback when it is applied not only to a keyword but to a word meaning representation, too.

To measure retrieval performance, we adopted the Mean-Average-Precision (MAP) and the Geometric-Mean-Average-Precision (GMAP) calculated by CLEF organizers using the DIRECT system on the basis of the first 1,000 retrieved items per request. Table 2 summarizes the description of system setup for each run, while Table 3 shows the results of five metrics (Mean-Average-Precision, Geometric-Mean-Average-Precision, R-precision, P@5 and P@10 are the precision after 5 and 10 documents retrieved respectively) for each run.

Table 3. Results of the performed experiments

Run	***MAP***	***GMAP***	***R-PREC***	***P@5***	***P@10***
unibaKTD	.3962	.1684	.3940	.4563	.3888
unibaKTDN	.4150	.1744	.4082	.4713	.4019
unibaKRF	.4250	.1793	.4128	.4825	.4150
unibaWsdTD	.2930	.1010	.2854	.3838	.3256
unibaWsdTDN	.3238	.1234	.3077	.4038	.3544
unibaWsdNL0802	.4218	.1893	.4032	.4838	.4081
unibaWsdNL0901	.4222	.1864	.4019	.4750	.4088
unibaKeySynRF	**.4346**	**.1960**	**.4153**	**.4975**	**.4188**
unibaCrossTD	.3414	.1131	.3389	.4013	.3419
unibaCrossTDN	.3731	.1281	.3700	.4363	.3713
unibaCrossKeyRF	**.3809**	.1311	**.3755**	.4413	.3794
unibaCrossWsdTD	.0925	.0024	.1029	.1188	.1081
unibaCrossWsdTDN	.0960	.0050	.1029	.1425	.1188
unibaCrossWsdNL0802	.3675	.1349	.3655	.4455	.3750
unibaCrossWsdNL0901	.3731	.1339	.3635	.4475	.3769
unibaCrossKeySynRF	.3753	**.1382**	.3709	**.4513**	**.3850**

Analyzing the mono-lingual task, the word meaning level used alone is not enough to reach good performance (*unibaWsdTD*, *unibaWsdTDN*). However, an increase of 1,7% in MAP is obtained when word meanings are exploited in the N-levels model (*unibaWsdNL0901*) with respect to the keyword level alone (*unibaKTDN*). Looking at the N-levels results, we can notice the impact of word meanings on GMAP. In fact, as the weight of the word meaning level raises the MAP decreases while the GMAP increases. In both runs, with or without WSD, the adoption of pseudo-relevance feedback techniques increases the MAP: 2.9% with WSD (*unibaKeySynRF* vs. *unibaWsdNL0901*) and 2.4% without WSD (*unibaKRF* vs. *unibaKTDN*). Finally, LCA combined to WSD (*unibaKeySynRF*) works better than LCA without WSD (*unibaKRF*) with an increment in all measures (+2.3% MAP, +9.3% GMAP, +0.6% R-prec, +3.1% P@5, +0.9% P@10) and, in general, it shows the best results.

In the bilingual task, queries are disambiguated using the first sense heuristics. This has an impact on the use of synsets in the query processing and pseudo-relevance feedback steps. Word meaning level performance is very bad. Moreover, runs without WSD generally outperform those with WSD, with an increment of

1.5% in MAP (*unibaCrossKeyRF* vs. *unibaCrossKeySynRF*). As LCA has shown to be helpful, with or without WSD, a higher increment is obtained without WSD: 2.09% in MAP (*unibaCrossKeyRF* vs. *unibaCrossTDN*). Nevertheless, also in the bilingual task WSD has improved the GMAP with an increment of 5.42% (*unibaCrossKeySynRF* vs. *unibaCrossKeyRF*). The increment in GMAP emphasizes the improvement for poorly performing (low precision) topics. This suggests that WSD is especially useful for those topics with low scores in average precision.

However, there are poorly performant queries (Average Precision<0.1). A query-by-query analysis suggested the reasons of the system failures. Hard topics can be split in two macro-categories. On one hand, there are complex topics which require precise information similar to a question answering task. For example, requests for game winners[4], name of countries/cities, events in specific periods of time (Topic 171, 172, 258, 310, 313, 345, 346). On the other hand, there are topics which specify non-relevance constraints (Topics 160, 305, 309[5], 322). Obviously, in the Bag-of-Word representation this kind of information is lost.

We validate our experiments (with respect to MAP metric) using both the parametric Student paired t-test and the non parametric Randomization test, as suggested in [6] ($\alpha = 5$ %). For the Randomization test we use a Perl script supplied by the authors. Both tests give similar results: all improvements are significant with only two exceptions. In both, mono/bi-lingual tasks without WSD, the differences obtained using the *NARRATIVE* field of the query are not significant. We achieve significant improvements in WSD task using the *NARRATIVE* field: this field is helpful to recognize the proper word meanings belonging to the query. In the monolingual task, the improvement obtained using the combination of keyword and word meaning levels are generally significant, except for *unibaWsdNL0802*.

6 Conclusion and Future Work

We have described and tested SENSE, a semantic N-levels IR system which manages documents indexed at multiple separate levels: keywords and meanings. The system is able to combine keyword search with semantic information provided by the other indexing levels.

With respect to the last participation of SENSE to CLEF 2008, we introduce in this edition new features in order to improve the overall retrieval performance. In particular, we adopt the Okapi BM25 model for both keyword and word meaning levels. Moreover, we propose a pseudo-relevance feedback strategy based on

[4] Who won a tennis Grand Slam Tournament event in 1995?

[5] What are the dangers to health of illegal hard drugs, such as heroin and cocaine, as opposed to soft drugs? Relevant documents must provide information on the medical risks involved in the illegal use of and dependence on hard drugs. Information on problems resulting from the use of "soft" drugs is not relevant.

Local Context Analysis. This strategy is applied to keyword and word meaning levels.

The results of the evaluation prove that the combination of keyword and word meaning can improve the retrieval performance. Only in bilingual task the combination of levels is outperformed by the only keyword level. Probably this is due to WSD technique adopted for Spanish topics. In particular, no WSD algorithms for Spanish are available and the organizers assign the first synset in Spanish-WordNet to each keyword in a topic. Moreover, the results prove that the pseudo-relevance feedback based on Local Context Analysis improves the IR performance.

References

1. Agirre, E., Di Nunzio, G.M., Ferro, N., Mandl, T., Otegi, A.: CLEF 2009 Ad Hoc Track Overview: Robust-WSD Task. In: Peters, C., Nunzio, G.D., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.) Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments. LNCS (LNAI), Springer, Heidelberg (2009)
2. Basile, P., Caputo, A., Gentile, A.L., Degemmis, M., Lops, P., Semeraro, G.: Enhancing Semantic Search using N-Levels Document Representation. In: Bloehdorn, S., Grobelnik, M., Mika, P., Tran, D.T. (eds.) SemSearch. CEUR Workshop Proceedings, vol. 334, pp. 29–43. CEUR-WS.org. (2008)
3. Fox, E.A., Shaw, J.A.: Combination of Multiple Searches. In: TREC, pp. 243–252 (1993)
4. Lee, J.H.: Analyses of multiple evidence combination. In: SIGIR, pp. 267–276. ACM, New York (1997)
5. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 extension to multiple weighted fields. In: CIKM 2004: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, pp. 42–49. ACM, New York (2004)
6. Smucker, M.D., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, pp. 623–632. ACM, New York (2007)
7. Sparck Jones, K., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments. Information Processing Management 36(6), 779–808, 809–840 2000)
8. Xu, J., Croft, W.B.: Improving the effectiveness of information retrieval with local context analysis. ACM Trans. Inf. Syst. 18(1), 79–112 (2000)

Using WordNet Relations and Semantic Classes in Information Retrieval Tasks

Javi Fernández, Rubén Izquierdo, and José M. Gómez

University of Alicante, Department of Software and Computing Systems
San Vicente del Raspeig Road, 03690 Alicante, Spain
javifm@ua.es, ruben@dlsi.ua.es, jmgomez@ua.es
http://www.dlsi.ua.es

Abstract. In this paper we explore the use of semantic classes in an existing information retrieval system in order to improve its results. Thus, we use two different ontologies of semantic classes (WordNet domain and Basic Level Concepts) in order to re-rank the retrieved documents and obtain better recall and precision. Finally, we implement a new method for weighting the expanded terms taking into account the weights of the original query terms and their relations in WordNet with respect to the new ones (which have demonstrated to improve the results). The evaluation of these approaches was carried out in the CLEF Robust-WSD Task, obtaining an improvement of 1.8% in GMAP for the semantic classes approach and 10% in MAP employing the WordNet term weighting approach.

1 Introduction

The two main goals of the Robust-WSD task are to measure the robustness of the retrieval systems (good stable performance over all queries) and test the benefits of the use of Word Sense Disambiguation (WSD) on this kind of systems. We decided to use an already implemented and evaluated system in last year's edition of CLEF as starting point for our approach. From all the available systems, we have chosen the Universidad Complutense de Madrid system [11], because of its good results, availability and the possibility of adapting the code easily to our objectives. Our main goal consists on experimenting the benefits of the use of *semantic classes* in Information Retrieval (IR) systems. However, we propose, also, a new and flexible way of weighting terms for the query expansion based on *WordNet relations*.

WSD, the task of assigning the correct sense to words depending on the context in which they appear, is a hard task and still a long way from being useful in other natural language processing applications, as shown in recent international evaluations [16,13]. The word senses these systems use are taken from a particular lexical semantic resource (most commonly WordNet [4].) WordNet has been widely criticized because of its too fine–grained sense distinctions, which are not useful for higher level applications like machine translation or question answering, and they are too subtle to be captured by automatic systems with the

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 158–165, 2010.

current small volumes of word–sense annotated examples. This can be a reason for the poor results of current WSD systems.

A possible solution is the use of *semantic classes* instead of word senses, because they group together senses of different words. This has several advantages: the average polysemy of texts is decreased, they provide richer and more useful information than word senses, and the amount of training data for each classifier is increased. Izquierdo et al. empirically explored the performnace of different levels of abstraction on the supervised WSD task [7]. These levels were provided by *WordNet Domains* (WND) [9], *SUMO labels* [10], *Lexicographer Files of WordNet* [4] and *Basic Level Concepts* (BLC20) [6]. Izquierdo et al. [7] referred to this approach as class–based WSD since the classifiers were created at a class level instead of at a sense level. As we have said, class–based WSD clusters senses of different words into the same explicit and comprehensive grouping. Only thoses cases belonging to the same semantic class are grouped to train the classifier. For example, the coarser word grouping obtained by Snow et al. [15] only has one remaining sense for "church". Using a set of Base Level Concepts [6], the three senses of "church" are still represented by *faith.n#3*, *building.n#1* and *religious_ceremony.n#1*.

We think that IR could take advantage with the use of word sense disambiguation, but from a semantic class point of view instead of the traditional word sense point of view. As to the data of the robust adhoc IR task has been processed automatically by two WSD systems, and the information of word senses is available, we did not run any class–based WSD system over the data. The next section describes the architecture of our system. In section 3 we discuss the results of this system at the CLEF 2009 Robust-WSD Task. Finally, in section 4 we draw conclusions and outline future works.

2 Description of the System

The system architecture is shown in Figure 1.

The user query is pre-parsed to obtain a set of terms without stopwords and any special symbol. Next, a ranked list of relevant documents are retrieved using the Lucene search engine[1]. With the retrieved documents, the initial query, the relations of the external resource WordNet and state-of-art query expansions methods an expanded query is obtained. The terms of this new query are weighted taking into account the weights of the original query terms, their relations in WordNet with respect to the new ones, the weight assigned by the WSD system to each sense and the weight returned by the expansion method. Once we have a new list of weighted terms, we perform another search but using the expanded query instead of the original one in order to retrieved a new ranked list of documents. Finally, we use the semantic class information from two different semantic resources (WordNet Domains and Base Level Concepts) in order to obtain a re-ranked document list as result.

In the following section we explain each of these processes in more detail.

[1] http://lucene.apache.org

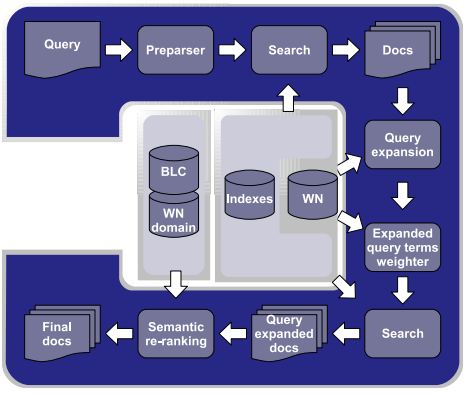


Fig. 1. Architecture of the system

2.1 Search Engine and Query Expansion

The search engine, which we are using, is the one provided by the Universidad Complutense de Madrid [11]. Their implementation is a modified version of Lucene which uses the BM25 probabilistic model [14] for document retrieval. They have also implemented two state-of-art query expansion methods: Kullback-Liebler Divergence [3] (an information-theoretic approach) and the Bo1 model [8,12] (based on Divergence From Randomness [2]). For our system we have chosen the Bo1 model because it is the approach with the best results in their evaluations. We also have decided to use the same constant values than they used in the last CLEF Robust-WSD edition in order to compare the effectiveness of our methods of semantic classes.

As we can see in Figure 1, we make two search processes. For the first retrieval process, query terms are lemmatized and stemmed in order to increase the system recall. The first *Search* module gets these terms as input and returns a list of relevant documents using the BM25 probabilistic model. Next, in the *Query expansion module*, we expand the original query obtaining new terms by means of the Bo1 model.

Although [11] proposed a method for weighting the expanded query terms based on WordNet, we have prefered to use our own method due to the fact that they do not use all senses of each term but the one with the highest weight. In our system we have decided to use all senses retrieved by the WSD system in order to improve the recall. In this way, the system searches all expanded terms in the relations of WordNet with respect to the synonyms, hyperonyms and hyponyms until a certain *distance*. For example, if the distance was 2, we search any expanded term among the hyperonym and hyponym synsets of the original terms but, also, the hyperonyms of the hyperonyms and the hyponyms of the hyponyms. The distance value sets the number of jumps to make in the WordNet relations from the synsets of the query terms. As we have mentioned above, we use all senses supplied for the WSD system for each query term but

we take into account the score given by these systems to each sense in order to calculate the weight of the expanded terms. Thefore, this distance factor is calculated by the following equation:

$$weight(synset_{i,d}) = weight(synset_{i,d-1}) * \alpha^d \quad (1)$$

We defined $synset_{i,1}$ as a given WordNet synset and $synset_{i,d}$ as another WordNet synset which is related to the $synset_{i,1}$ of a distance of d jumps (taken into acount only hyperonym and hyponym relations). Thus, $weight(synset_{i,d})$ is the weight of the synset i, d and $weight(synset_{i,1})$ is the score given by the WSD system to the synset i, d. α is a constant whose value is between 0 and 1 and d the distance of $synset_{i,d}$ to the $synset_{i,1}$.

Once we have calculated the previous synset weight, we combine this weight with the weight assigned by the expanded method bo1 in order to calculate the final term weight using the following equation:

$$weight(term_t) = \frac{weight(synset_{i,d}) + weight_0(term_t)}{2} \quad (2)$$

Where $weight(term_t)$ is the weight of the expanded term t which is grouped in the WordNet synset i, d, and $weight_0(term_t)$ is the weight assigned by bo1 to the term t.

With these equations, we give importance to those expanded terms which are related to more likely the original query terms and, in addition, we include the score given by the WSD system for each query term in the final term weight. Thus, we include all senses of a term in the search but we give more importance to those terms which are related to more likely senses and closer to the original query terms.

2.2 Semantic Classes

Our approach consists in mapping the assigned word senses to semantic classes, specifically to WordNet Domains labels and Basic Level Concepts.

WordNet Domains [9] is a hierarchy of 165 Domain Labels which have been used to label all the WordNet synsets. Information brought by Domain Labels is complementary to what is already in WordNet. First of all, a Domain Label can include synsets of different syntactic categories: for instance MEDICINE groups together senses from nouns, such as doctor or hospital, and from verbs, such as to operate. Second, a Domain Label may also contains senses from different WordNet subhierarchies. For example, SPORT contains senses such as athlete, deriving from life form, game equipment from physical object, sport from act and playing field from location.

Basic Level Concepts [6] are a set of concepts that result from the compromise between two conflicting principles of characterization: represent as many concepts as possible and represent as many features as possible. As a result of this, Basic Level Concepts typically occur in the middle of hierarchies and less frequently than the maximum number of relations.

The authors developed a method for the automatic selection of BLC from WordNet. They use a very simple method for deriving a small set of appropriate meanings using basic structural properties of WordNet. The approach considers:

- The total number of relations of every synset or just the hyponymy relations.
- Discard those BLCs that do not represent at least a number of synsets.
- Optionally, the frequency of the synsets (summing up the frequency of the senses provided by WordNet).

The process of automatic selection of BLC follows a bottom-up approach using the chain of hypernym relations. For each synset in WN, the process selects as its Base Level Concept the first local maximum according to the relative number of relations. For synsets having multiple hypernyms, the path having the local maximum with higher number of relations is selected. Usually, this process finishes having a number of false Base Level Concepts. That is, synsets having no descendants (or with a very small number) but being the first local maximum according to the number of relations considered. Thus, the process finishes by checking if the number of concepts subsumed by the preliminary list of BLC is higher than a certain threshold. For those BLC not representing enough concepts according to a certain threshold, the process selects the next local maximum following the hypernym hierarchy.

Thus, depending on the type of relations considered to be counted and the threshold established, different sets of BLC can be easily obtained for each WN version. For our work, we have selected the set of BLC built using all kind of relations and a threshold of 20 as the minimum number of synsets that each BLC must subsume.

We explain now the **representation of documents or queries with semantic classes** of the words contained on them. In the task data, each ambiguous word is annotated with its possible senses, each one with a certain probability. Starting from this information, we create a domain vector, for a query or for a document, containing all the semantic classes information of the query or document. The Domain vector consists of a vector where each element represents a WordNet Domain or a Basic Level Concept and its associated weight. Note that there are 165 Domain Labels in WordNet Domains and 558 Basic Level Concepts for nouns. The way to build this vector is: each word has annotated several senses, with the associated probability; each word sense is mapped to its proper semantic class, and the element of the vector corresponding to this Domain is increased with the probability associated with the word sense. When all the words are processed, we obtain a domain vector representing the semantic information of the document or query. Finally to compare two documents, or a document and a query, and obtain their similarity in terms of their semantic content, we use the value of the cosine defined by the two domain vectors.

2.3 Integration of Semantic Classes in Robust Ad Hoc

Once the final list of documents from the expanded query is retrieved, the *Semantic re-ranking* module rearranges this list taking into account both the similarity

returned by the BM25 probabilistic model and the similarity calculated by the semantic class system. In order to do this, we have studied several equations described in [5]. In this paper, the best results were the ones obtained with the following equation:

$$semsim(i,j) = \begin{cases} simmax_i + sim_{ij} * sem_{ij} & \text{if } sem_{ij} > h \\ sim_{ij} & \text{otherwise} \end{cases} \quad (3)$$

Where $semsim(i,j)$ is the final similarity between the query i and the document j, sim_{ij} is the similarity of the query i with respect to the document j returned by the search engine, sem_{ij} is the same similarity but returned by the semantic class system, $simmax_i$ is the greatest value of similarity returned by the search engine for the query i and h is a constant which determines a semantic similarity threshold defined empirically. This equation gives more relevance those documents with high semantic similarity but takes into account the semantic class score in the final similarity value.

3 Evaluation

In this section we report the results of each one of our proposals separately.

For the evaluation of the *Expanded query terms weighter*, we have to set the value for two variables: α and d (distance). In order to get the best values for these variables, we have experimented with several different values for them. In table 1 we present two of the best results of those experiments. With $\alpha = 0.8$ and $d = 1$ we improve the baseline GMAP in a 9.97%. With $\alpha = 0.92$ and $d = 6$ we improve both the baseline MAP in a 0.02% and the baseline GMAP in a 8.19%.

For the evaluation of the *Semantic re-ranking*, the only variable is the threshold h for the reranker. We have experimented with different values for this variable in order to obtain the best results. In Table 2 we present the best results

Table 1. Evaluation of the *Expanded query terms weighting* module

	MAP	GMAP	R-Prec	P@5	P@10
BM25 + Bo1 (Baseline)	.3737	.1294	.0.3585	.4475	**.3825**
BM25 + Bo1 + WD ($\alpha = 0.8$, $d = 1$)	.3706	**.1423**	.3624	.4500	.3750
BM25 + Bo1 + WD ($\alpha = 0.92$, $d = 6$)	**.3738**	.1400	**.3655**	**.4513**	.3775

Table 2. Evaluation of the *Semantic re-ranking* module

	MAP	GMAP	R-Prec	P@5	P@10
BM25 + Bo1 (Baseline)	.3737	.1294	.3585	**.4475**	.3825
BM25 + Bo1 + WND + RR ($h = 0.5$)	.3752	.1298	**.3638**	.4462	**.3862**
BM25 + Bo1 + BLC20 + RR ($h = 0.8$)	**.3776**	**.1317**	.3609	.4437	.3806

of those experiments for each semantic classes model. The integration of the semantic classes to the search engine improves the baseline results. With WND we improve both the baseline MAP in a 0.4% and the baseline GMAP in a 0.31%. With BLC20 we improve both the baseline MAP in a 0.64% and the baseline GMAP in a 1.77%.

4 Conclusions

The results of the experiments with our two proposals have shown improvements to the initial information retrieval system.

In the first one, the *Expanded query terms weighter* module, we have experimented with the weights of the terms in a probabilistic IR system. We have applied a smoothing function based on the WordNet distance to the weights given by the IR system. The experiments made have shown GMAP improvements of nearly 10% but not significant MAP improvements.

As future work we propose to continue with the experiments on this module. For the propagation function 2, the search of the best values for α and d can be more exhaustive, finding better values for this variables. Moreover, new relations can be explored in WordNet (not only hyponyms and hyperonyms), in order to improve recall. Even new weight propagation functions can be proposed to better exploit the concept of *distance* in WordNet.

In the second of our proposals, the *Semantic re-ranking* module, we have integrated the semantic classes to a IR system. We have done this integration recalculating the weight of the documents retrieved depending on the similarity between the semantic class of each document and the semantic class of the query. The results of the experiments made reveal that the semantic classes resources can be effectively integrated to the IR systems.

This module can also be applied at new levels. We only have used five simple integration functions for the search engine and the semantic classes weights. More functions can be studied to find the best way to integrate the available resources of semantic classes.

Acknowledgements

This paper has been partially supported by the Spanish government, project TIN-2006-15265-C06-01 and by the framework of the project QALL-ME, which is a 6th Framework Research Programme of the European Union (EU), contract number: FP6-IST-033860 and by the University of Comahue under the project 04/E062.

References

1. Aguirre, E., Di Nunzio, G.M., Mandl, T., Otegi, A.: Clef 2009 ad hoc track overview: Robust-wsd task. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I, Corfou, Greece. LNCS, vol. 6241,pp. 36–49. Springer, Heidelberg (2010)

2. Amati, G., Van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. 20(4), 357–389 (2002)
3. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley-Interscience, New York (1991)
4. Fellbaum, C. (ed.): WordNet. An Electronic Lexical Database. The MIT Press, Cambridge (1998)
5. Fernández, J., Izquierdo, R., Gómez, J.M.: Alicante at clef 2009 robust-wsd task. In: Working notes of Cross Language Evaluation Forum 2008, Corfou, Greece, CLEF (2009)
6. Izquierdo, R., Suárez, A., Rigau, G.: Exploring the automatic selection of basic level concepts. In: Angelova, G., et al. (eds.) International Conference Recent Advances in Natural Language Processing, Borovets, Bulgaria, pp. 298–302 (2007)
7. Izquierdo, R., Suárez, A., Rigau, G.: An empirical study on class-based word sense disambiguation. In: EACL, pp. 389–397. The Association for Computer Linguistics (2009)
8. Macdonald, C., He, B., Plachouras, V., Ounis, I.: University of glasgow at trec 2005: Experiments in terabyte and enterprise tracks with terrier. In: TREC (2005)
9. Magnini, B., Cavaglià, G.: Integrating subject field codes into wordnet. In: Proceedings of LREC, Athens, Greece (2000)
10. Niles, I., Pease, A.: Towards a standard upper ontology. In: Weltyand, C., Smith, B. (eds.) Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS 2001), pp. 17–19 (2001)
11. Peréz Agüera, J.R., Zaragoza, H.: Query clauses and term independence. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 138–145. Springer, Heidelberg (2009)
12. Plachouras, V., He, B., Ounis, I.: University of glasgow at trec 2004: Experiments in web, robust, and terabyte tracks with terrier. In: Voorhees, E.M., Buckland, L.P., Voorhees, E.M., Buckland, L.P. (eds.) TREC, NIST (2004)
13. Pradhan, S., Loper, E., Dligach, D., Palmer, M.: Semeval-2007 task-17: English lexical sample, srl and all words. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval 2007), Prague, Czech Republic, June 2007, pp. 87–92. Association for Computational Linguistics (June 2007)
14. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR 1994: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 232–241. Association for Computational Linguistics (1994)
15. Snow, R., Prakash, S., Jurafsky, D., Ng, A.: Learning to merge word senses. In: Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 1005–1014 (2007)
16. Snyder, B., Palmer, M.: The english all-words task. In: Mihalcea, R., Edmonds, P. (eds.) Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, pp. 41–43. Association for Computational Linguistics (July 2004)

Using Semantic Relatedness and Word Sense Disambiguation for (CL)IR

Eneko Agirre[1], Arantxa Otegi[1], and Hugo Zaragoza[2]

[1] IXA NLP Group - University of Basque Country. Donostia, Basque Country
{e.agirre,arantza.otegi}@ehu.es
[2] Yahoo! Research, Barcelona, Spain
hugoz@yahoo-inc.com

Abstract. In this paper we report the experiments for the CLEF 2009 Robust-WSD task, both for the monolingual (English) and the bilingual (Spanish to English) subtasks. Our main experimentation strategy consisted of expanding and translating the documents, based on the related concepts of the documents. For that purpose we applied a state-of-the art semantic relatedness method based on WordNet. The relatedness measure was used with and without WSD information. Even though we obtained positive results in our training and development datasets, we did not manage to improve over the baseline in the monolingual case. The improvement over the baseline in the bilingual case is marginal. We plan further work on this technique, which has attained positive results in the passage retrieval for question answering task at CLEF (ResPubliQA).

1 Introduction

Our goal is to test whether Word Sense Disambiguation (WSD) information can be beneficial for Cross Lingual Information Retrieval (CLIR) or monolingual Information Retrieval (IR). WordNet has been previously used to expand the terms in the query with some success [5] [6] [7] [9]. WordNet-based approaches need to deal with ambiguity, which proves difficult given the little context available to disambiguate the words in the query effectively. In our experience document expansion works better than topic expansion (see our results for the previous edition of CLEF in [8]). Bearing this in mind, in this edition we have mainly focused on documents, using a more elaborate expansion strategy. We have applied a state-of-the-art semantic relatedness method based on WordNet [3] in order to select the best terms to expand the documents. The relatedness method can optionally use the WSD information provided by the organizers.

The remainder of this paper is organized as follows. Section 2 describes the experiments carried out. Section 3 presents the results obtained and Section 4 analyzes the results. Finally, Section 5 draws conclusions and mentions future work.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 166–173, 2010.

2 Experiments

Our main experimentation strategy consisted of expanding the documents, based on the related concepts of the documents. The steps of our retrieval system are the following. We first expand/translate the topics. In a second step we extract the related concepts of the documents, and expand the documents with the words linked to these concepts in WordNet. Then we index these new expanded documents, and finally, we search for the queries in the indexes in various combinations. All steps are described sequentially.

2.1 Expansion and Translation Strategies of the Topics

WSD data provided to the participants was based on WordNet version 1.6. In the topics each word sense has a WordNet synset assigned with a score. Using those synset codes and the English and Spanish wordnets, we expanded the topics. In this way, we generated different topic collections using different approaches of expansion and translation, as follows:

- Full expansion of English topics: expansion to all synonyms of all senses.
- Best expansion of English topics: expansion to the synonyms of the sense with highest WSD score for each word, using either UBC or NUS disambiguation data (as provided by organizers).
- Translation of Spanish topics: translation from Spanish to English of the first sense for each word, taking the English variants from WordNet.

In both cases we used the Spanish and English wordnet versions provided by the organizers.

2.2 Query Construction

We constructed queries using the title and description topic fields. Based on the training topics, we excluded some words and phrases from the queries, such as *find, describing, discussing, document, report* for English and *encontrar, describir, documentos, noticias, ejemplos* for Spanish.

After excluding those words and taking only nouns, adjectives, verbs and numbers, we constructed several queries for each topic using the different expansions of the topics (see Section 2.1) as follows:

- Original words.
- Both original words and expansions for the best sense of each word.
- Both original words and all expansions for each word.
- Translated words, using translations for the best sense of each word. If a word had no translation, the original word was included in the query.

The first three cases are for the monolingual runs, and the last one for the bilingual run which translated the query.

2.3 Expansion and Translation Strategies of the Documents

Our document expansion strategy was based on semantic relatedness. For that purpose we used UKB[1], a collection of programs for performing graph-based Word Sense Disambiguation and lexical similarity/relatedness using a pre-existing knowledge base, in this case WordNet 1.6.

Given a document, UKB returns a vector of scores for each concept in WordNet. The higher the score, the more related is the concept to the given document. In our experiments we used different approaches to represent each document:

- using all the synsets of each word of the document.
- using only the synset with highest WSD score for each word, as given by the UBC disambiguation data [2] (provided by the organizers).

In both cases, UKB was initialized using the WSD weights: each synset was weighted with the score returned by the disambiguation system, that is, each concept was weighted according to the WSD weight of the corresponding sense of the target word.

Once UKB outputs the list of related concepts, we took the highest-scoring 100 or 500 concepts and expanded them to all variants (words in the concept) as given by WordNet. For the bilingual run, we took the Spanish variants. In both cases we used the Spanish and English wordnet versions provided by the organizers.

The variants for those expanded concepts were included in two new fields of the document representation; 100 concepts in the first field and 400 concepts in the second field. This way, we were able to use the original words only, or also the most related 100 concepts, or the original words and the most related 500 concepts. We will get back to this in Section 2.4 and Section 2.5.

Figure 2 shows a document expansion for the document in Figure 1. The second column in Figure 2 is the vector of related concepts (synsets values) returned by UKB for the mentioned document. The vector in the example is sorted by the score for each concept (first column). So the concepts that are shown on it are the most related concepts for that document. The words in the third column are the variants for each concept taken from WordNet. We also added these words to another index. The terms in bold in the example are the words that appear in the document. And the terms in italic are the new terms that we obtain by means of the expansion.

2.4 Indexing

We indexed the new expanded documents using the MG4J search-engine [4]. MG4J makes it possible to combine several indices over the same document collection. We created one index for each field: one for the original words, one for the expansion of the top 100 concepts, and another one for the expansion of the following 400 concepts. The Porter stemmer was used with default settings.

[1] The algorithm is publicly available at http://ixa2.si.ehu.es/ukb/

```
HUNTINGTON BANK ROBBERY NETS $780
A man walked into a bank Friday, warned a teller that he had a gun and
made off with $780, police said.
Huntington Beach Police Sgt. Larry Miller said the teller at the World
Savings and Loan Assn., 6902 Warner Ave., did not see a weapon during
the robbery, which occurred at 4:35 p.m.
The robber escaped out the west door of the building. Police have no
suspects in the case.
```

Fig. 1. Document example

0.0071192807	06093563−*n* ⟹	*constabulary*, *law*, **police**, *police force*
0.007016694	02347413−*n* ⟹	**building**, *edifice*
0.00701617062	07635368−*n* ⟹	**teller**, *vote counter*
0.00700878272	06646591−*n* ⟹	**huntington**
0.0070066648	00499726−*n* ⟹	**robbery**
0.006932565	00235191−*v* ⟹	*come about*, *go on*, *hap*, *happen*, **occur**, *pass*, *pass off*, *take place*
0.006929787	03601056−*n* ⟹	*arm*, **weapon**, *weapon system*
0.006903118	01299603−*v* ⟹	**walk**
0.006898292	02588950−*n* ⟹	**door**
0.006894822	02778084−*n* ⟹	**gun**
0.006892254	09651550−*n* ⟹	**loan**
0.0068790509	06739108−*n* ⟹	**beach**
0.0068660484	10937709−*n* ⟹	**p.m.**, *pm*, *post meridiem*
0.006831742	10883362−*n* ⟹	*fri*, **friday**
0.0068182234	07422992−*n* ⟹	*mugger*, **robber**
0.00676897472	07410610−*n* ⟹	**miller**
0.0058595173	00126393−*n* ⟹	*economy*, **saving**
0.0055009496	00465486−*v* ⟹	**suspect**
0.0053402969	00589833−*v* ⟹	**warn**
0.005200375	07391044−*n* ⟹	*adult male*, **man**
...	...	...

Fig. 2. Example for an expansion

2.5 Retrieval

We carried out several retrieval experiments combining different kind of queries with different kind of indices. We used the training data to perform extensive experimentation, and chose the ones with best MAP results in order to produce the test topic runs.

The different kind of queries that we had prepared are those explained in Section 2.2. Our experiments showed that original words were getting good results, so in the test runs we used only the queries with original words.

MG4J allows multi-index queries, where one can specify which of the indices one wants to search in, and assign different weights to each index. We conducted

different experiments, by using the original words alone (the index made of original words) and also by using one or both indices with the expansion of concepts, giving different weight to the original words and the expanded concepts. The best weights were then used in the test set, as explained in the following Section.

We used the BM25 ranking function with the following parameters: 1.0 for *k1* and 0.6 for *b*. We did not tune these parameters.

The submitted runs are described in Section 3.

3 Results

Table 1 summarizes the results of our submitted runs. The IR process is the same for all the runs and the main differences between them is the expansion strategy. The characteristics of each run are as follows:

- monolingual without WSD:
  - **EnEnNowsd**: original terms in topics; original terms in documents.
- monolingual with WSD:
  - **EnEnAllSenses100Docs**: original terms in topics; both original and expanded terms of 100 concepts, using all senses for initializing the semantic graph. The weight of the index that included the expanded terms: 0.25.
  - **EnEnBestSense100Docs**: original terms in topics; both original and expanded terms of 100 concepts, using best sense for initializing the semantic graph. The weight of the index that included the expanded terms: 0.25.
  - **EnEnBestSense500Docs**: original terms in topics; both original and expanded terms of 500 concepts, using best sense for initializing the semantic graph. The weight of the index that included the expanded terms: 0.25.
- bilingual without WSD:
  - **EsEnNowsd**: translated terms in topics (from Spanish to English); original terms in documents (in English).
- bilingual with WSD:
  - **EsEn1stTopsAllSenses100Docs**: translated terms in topics (from Spanish to English); both original and expanded terms of 100 concepts, using all senses for initializing the semantic graph. The weight of the index that included the expanded terms: 0.15.
  - **EsEn1stTopsBestSense500Docs**: translated terms in topics (from Spanish to English); both original and expanded terms of 100 concepts, using best sense for initializing the semantic graph. The weight of the index that included the expanded terms: 0.15.
  - **EsEnAllSenses100Docs**: original terms in topics (in Spanish); both original terms (in English) and translated terms (in Spanish) in documents, using all senses for initializing the semantic graph. The weight of the index that included the expanded terms: 1.00.

- **EsEnBestSense500Docs**: original terms in topics (in Spanish); both original terms (in English) and translated terms (in Spanish) in documents, using best sense for initializing the semantic graph. The weight of the index that included the expanded terms: 1.60.

The weight of the index which was created using the original terms of the documents was 1.00 for all the runs.

Table 1. Results for submitted runs

		runId	map	gmap
monolingual	no WSD	EnEnNowsd	**0.3826**	0.1707
	with WSD	EnEnAllSenses100Docs	0.3654	0.1573
		EnEnBestSense100Docs	0.3668	0.1589
		EnEnBestSense500Docs	**0.3805**	0.1657
bilingual	no WSD	EsEnNowsd	**0.1805**	0.0190
	with WSD	EsEn1stTopsAllSenses100Docs	0.1827	0.0193
		EsEn1stTopsBestSense500Docs	**0.1838**	0.0198
		EsEnAllSenses100Docs	0.1402	0.0086
		EsEnBestSense500Docs	0.1772	0.0132

Regarding monolingual results, we can see that using the best sense for representing the document when initializing the semantic graph achieves slightly higher results with respect to using all senses. Besides, we obtained better results when we expanded the documents using 500 concepts than using only 100 (compare the results of the runs `EnEnBestSense100Docs` and `EnEnBestSense500Docs`). However, we did not achieve any improvement over the baseline with either WSD or semantic relatedness information. We have to mention that we did achieve improvement in the training data, but the difference was not significant[2].

With respect to the bilingual results, `EsEn1stTopsBestSense500Docs` obtains the best result, although the difference with respect to the baseline run is not statistically significant. This is different to the results obtained using the training data, where the improvements using the semantic expansion were remarkable (4.91% of improvement over MAP). It is not very clear whether translating the topics from Spanish to English or translating the documents from English to Spanish is better, since we got better results in the first case in the testing phase (see runs called `...1stTops...` in the Table 1), but not in the training phase.

In our experiments we did not make any effort to deal with hard topics, and we only paid attention to improvements in Mean Average Precision (MAP) metric. In fact, we applied the settings which proved best in training data according to MAP. Another option could have been to optimize the parameters and settings according to Geometric Mean Average Precision (GMAP) values.

[2] We used paired Randomization Tests over MAPs with α=0.05.

4 Analysis

In this section we focus on comparison, on the one hand, between different approaches of using WSD data for IR, and on the other hand, between different collections used to test the document expansion strategies for IR.

The expansion strategy we used in the previous edition of the task consisted of expanding documents with synonyms based on WSD data and it provided consistent improvements over the baseline, both in monolingual and bilingual tasks [8]. With the document expansion strategy presented in this paper we achieve gains over the baseline in monolingual task using training data and in bilingual task both in training and testing phases.

With respect to using different datasets, we found that using semantic relatedness to expand documents can be effective for the passage retrieval task (ResPubliQA) [1]. The strategy used in it differs from the one explained here, as the expansion is done using the variants of the synsets, rather than the synsets themselves. After the competition, we applied this expansion strategy to the dataset of the Robust task and the monolingual results raised up to 0.3875.

5 Conclusions and Future Work

We have described our experiments and the results obtained in both monolingual and bilingual tasks at Robust-WSD Track at CLEF 2009. Our main experimentation strategy consisted of expanding the documents based on a semantic relatedness algorithm.

The objective of carrying out different expansion strategies was to study if WSD information and semantic relatedness could be used in an effective way in (CL)IR. After analyzing the results, we have found that those expansion strategies were not very helpful, especially in the monolingual task.

For the future, we want to analyze expansion using variants of the related concepts, as it attained remarkable improvements in the passage retrieval task (ResPubliQA) [1].

Acknowledgments

This work has been supported by KNOW2 (TIN2009-14715-C04-01) and KYOTO (ICT-2007-211423). Arantxa Otegi's work is funded by a PhD grant from the Basque Government. Part of this work was done while Arantxa Otegi was visiting Yahoo! Research Barcelona.

References

1. Agirre, E., Ansa, O., Arregi, X., Lopez de Lacalle, M., Otegi, A., Saralegi, X., Zaragoza, H.: Elhuyar-IXA: Semantic Relatedness and Cross-Lingual Passage Retrieval. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 273–280. Springer, Heidelberg (2010)

2. Agirre, E., Lopez de Lacalle, O.: UBC-ALM: Combining k-NN with SVD for WSD. In: Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007), Czech Republic, Prague, pp. 341–345 (2007)
3. Agirre, E., Soroa, A., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M.: A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In: Proceedings of Annual Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL), Boulder, USA (2009)
4. Boldi, P., Vigna, S.: MG4J at TREC 2005. The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings, NIST Special Publications, SP 500-266 (2005), http://mg4j.dsi.unimi.it/
5. Kim, S., Seo, H., Rim, H.: Information Retrieval using word senses: Root sense tagging approach. In: Proceedings of SIGIR 2004 (2004)
6. Liu, S., Liu, F., Yu, C., Meng, W.: An effective approach to document retrieval via utilizing WordNet and recognizing phrases. In: Proceedings of SIGIR 2004 (2004)
7. Liu, S., Yu, C., Meng, W.: Word Sense Disambiguation in Queries. In: Proceedings of ACM Conference on Information and Knowledge Management, CIKM (2005)
8. Otegi, A., Agirre, E., Rigau, G.: IXA at CLEF 2008 Robust-WSD Task: using Word Sense Disambiguation for (Cross Lingual) Information Retrieval. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 118–125. Springer, Heidelberg (2009)
9. Pérez-Agüera, J.R., Zaragoza, H.: Query Clauses and Term Independence. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 138–145. Springer, Heidelberg (2009)

Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation

Anselmo Peñas[1], Pamela Forner[2], Richard Sutcliffe[3], Álvaro Rodrigo[4], Corina Forăscu[5], Iñaki Alegria[6], Danilo Giampiccolo[7], Nicolas Moreau[8], and Petya Osenova[9]

[1] UNED, Spain
anselmo@lsi.uned.es
[2] CELCT, Italy
forner@celct.it
[3] DLTG University of Limerick, Ireland
richard.sutcliffe@ul.ie
[4] UNED, Spain
alvarory@lsi.uned.es
[5] UAIC and RACAI, Romania
corinfor@info.uaic.ro
[6] University of Basque Country, Spain
i.alegria@ehu.es
[7] CELCT, Italy
giampiccolo@celct.it
[8] ELDA/ELRA, France
moreau@elda.org
[9] BTB, Bulgaria
petya@bultreebank.org

Abstract. This paper describes the first round of ResPubliQA, a Question Answering (QA) evaluation task over European legislation, proposed at the Cross Language Evaluation Forum (CLEF) 2009. The exercise consists of extracting a relevant paragraph of text that satisfies completely the information need expressed by a natural language question. The general goals of this exercise are (i) to study if the current QA technologies tuned for newswire collections and Wikipedia can be adapted to a new domain (law in this case); (ii) to move to a more realistic scenario, considering people close to law as users, and paragraphs as system output; (iii) to compare current QA technologies with pure Information Retrieval (IR) approaches; and (iv) to introduce in QA systems the Answer Validation technologies developed in the past three years. The paper describes the task in more detail, presenting the different types of questions, the methodology for the creation of the test sets and the new evaluation measure, and analyzing the results obtained by systems and the more successful approaches. Eleven groups participated with 28 runs. In addition, we evaluated 16 baseline runs (2 per language) based only in pure IR approach, for comparison purposes. Considering accuracy, scores were generally higher than in previous QA campaigns.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 174–196, 2010.

1 Introduction

This year, the Multilingual Question Answering Track proposed three separate and independent exercises:

1. *QAST*: The aim of the third QAST exercise is to evaluate QA technology in a real multilingual speech scenario in which written and oral questions (factual and definitional) in different languages are formulated against a set of audio recordings related to speech events in those languages. The scenario is the European Parliament sessions in English, Spanish and French.
2. *GikiCLEF*: Following the previous GikiP pilot at GeoCLEF 2008, the task focuses on open list questions over Wikipedia that require geographic reasoning, complex information extraction, and cross-lingual processing, at least for Dutch, English, German, Norwegian, Portuguese and Romanian.
3. *ResPubliQA*: Given a pool of 500 independent questions in natural language, systems must return the passage - not the exact answer - that answers each question. The document collection is JRC-Acquis about EU documentation[1]. Both questions and documents are translated into and aligned for a subset of official European languages, i.e. Bulgarian, English, French, German, Italian, Portuguese, Romanian and Spanish.

This overview is dedicated only to the ResPubliQA exercise. For more details about QAST and GikiCLEF see the respective overviews in this volume.

The ResPubliQA 2009 exercise is aimed at retrieving answers to a set of 500 questions. The answer of a question is a paragraph of the test collection. The hypothetical user considered for this exercise is a person interested in making inquiries in the law domain, specifically on the European legislation. The ResPubliQA document collection is a subset of JRC-Acquis[1], a corpus of European legislation that has parallel translations aligned at document level in many European languages.

In the ResPubliQA 2009 exercise, participating systems could perform the task in any of the following languages: Basque (EU), Bulgarian (BG), English (EN), French (FR), German (DE), Italian (IT), Portuguese (PT), Romanian (RO) and Spanish (ES). All the monolingual and bilingual combinations of questions between the languages above were activated, including the monolingual English (EN) task – usually not proposed in the QA track at CLEF. Basque (EU) was included exclusively as a source language, as there is no Basque collection available - which means that no monolingual EU-EU sub-task could be enacted.

The paper is organized as follows: Section 2 gives an explanation of the task objectives; Section 3 illustrates the document collection; Section 4 gives an overview of the different types of question developed; Section 5 addresses the various steps to create the ResPubliQA data set; Section 6 shows the format of the test set and of the submissions that systems have returned; Section 7 provides an explanation of the evaluation measure and of how systems have been evaluated; Section 8 gives some details about participation in this year evaluation campaign; Section 9 presents and discusses the results achieved by participating systems and across the different languages; Section 10 shows

[1] http://wt.jrc.it/lt/Acquis/

the methodologies and technique used by participating systems and Section 11 and 12 draws some conclusions highlighting the challenges which are still to be addressed.

2 Task Objectives

The general objectives of the exercise are:

1. Moving towards a domain of potential users. While looking for a suitable context, improving the efficacy of legal searches in the real world seemed an approachable field of study. The retrieval of information from legal texts is an issue of increasing importance given the vast amount of data which has become available in electronic form over the last few years.

Moreover, the legal community has showed much interest in IR technologies as it has increasingly faced the necessity of searching and retrieving more and more accurate information from large heterogeneous electronic data collections with a minimum of wasted effort.

In confirmation of the increasing importance of this issue, a Legal Track [14], aimed at advancing computer technologies for searching electronic legal records, was also introduced in 2006 as part of the yearly TREC conferences sponsored by the National Institute of Standards and Technology (NIST).[2] The task of the Legal Track is to retrieve all the relevant documents for a specific query and to compare the performances of systems operating in a setting which reflects the way lawyers carry out their inquiries.

2. Studying if current QA technologies tuned for newswire collections and Wikipedia can be easily adapted to a new domain (law domain in this case). It is not clear if systems with good performance in newswire collections, after many years spent adapting the system to the same collections, perform well in a new domain. In this sense, the task is a new challenge for both, seniors and newcomers.

3. Moving to an evaluation setting able to compare systems working in different languages. Apart from the issue of domain, a shortcoming of previous QA campaigns at CLEF was that each target language used a different document collection. This meant that the questions for each language had to be different and as a consequence the performance of systems was not directly comparable unless they happened to work with the same target language.

In the current campaign, this issue was addressed by adopting a document collection which is parallel at the document level in all the supported languages. This meant that for the first time, all participating systems were answering the same set of questions even though they might be using different languages.

4. Comparing current QA technologies with pure Information Retrieval (IR) approaches. Returning a complete paragraph instead of an exact answer allows the comparison between pure IR approaches and current QA technologies. In this way, a nice benchmark for evaluating IR systems oriented to high precision, where only one paragraph is needed, has been also created. The documents are nicely divided into

[2] It may be interesting to know that in 2008 the TREC QA Track moved to the Text Analysis Conference (TAC). In 2009 no QA Track has been proposed at any conferences sponsored by NIST.

xml paragraph marks solving the technical issues for paragraph retrieval. Furthermore, a paragraph is presumably a more realistic output for the users of the new collection domain.

5. Allowing more types of questions. Returning one paragraph allows new types of questions with the only restriction that they must be answered by a single paragraph.

6. Introducing in QA systems the Answer Validation technologies developed in the past campaigns. During the last campaigns we wanted to stick to the easiest and most comprehensible evaluation of systems, that is, requesting only one answer per question and counting the proportion of questions correctly answered (namely accuracy). In this campaign, we wanted to introduce a more discriminative measure, allowing systems to leave some questions unanswered. Given two systems that answer correctly the same proportion of questions, the one that returns less incorrect answers (leaving some questions unanswered) will score better. Thus, systems can add a final module to decide whether they found enough evidence or not to return their best answer.

This is a classification problem that takes advantage of more sophisticated Answer Validation technologies developed during the last years [8,9,12].

3 Document Collection

The ResPubliQA collection is a subset of the JRC-ACQUIS Multilingual Parallel Corpus[3]. JRC-Acquis is a freely available parallel corpus containing the total body of European Union (EU) documents, mostly of legal nature. It comprises contents, principles and political objectives of the EU treaties; the EU legislation; declarations and resolutions; international agreements; and acts and common objectives. Texts cover various subject domains, including economy, health, information technology, law, agriculture, food, politics and more. This collection of legislative documents currently includes selected texts written between 1950 and 2006 with parallel translations in 22 languages. The corpus is encoded in XML, according to the TEI guidelines.

The ResPubliQA collection in 8 of the languages involved in the track - Bulgarian, English, French, German, Italian, Portuguese, Romanian and Spanish - consists of roughly 10,700 parallel and aligned documents per language. The documents are grouped by language, and inside each language directory, documents are grouped by year. All documents have a numerical identifier called the CELEX code, which helps to find the same text in the various languages. Each document contains a header (giving for instance the download URL and the EUROVOC codes) and a text (which consists of a title and a series of paragraphs).

4 Types of Questions

The questions fall into the following categories: Factoid, Definition, Reason, Purpose, and Procedure.

[3] Please note that it cannot be guaranteed that a document available on-line exactly reproduces an officially adopted text. Only European Union legislation published in paper editions of the Official Journal of the European Union is deemed authentic.

Factoid. Factoid questions are fact-based questions, asking for the name of a person, a location, the extent of something, the day on which something happened, etc. For example:

Q: *When must animals undergo ante mortem inspection?*
A: 9. Animals must undergo ante mortem inspection on the day of their arrival at the slaughterhouse. The inspection must be repeated immediately before slaughter if the animal has been in the lairage for more than twenty-four hours.

Q: *In how many languages is the Official Journal of the Community published?*
A: The Official Journal of the Community shall be published in the four official languages.

Definition. Definition questions are questions such as "What/Who is X?", i.e. questions asking for the role/job/important information about someone, or questions asking for the mission/full name/important information about an organization. For example:

Q: *What is meant by "whole milk"?*
A: 3. For the purposes of this Regulation, 'whole milk' means the product which is obtained by milking one or more cows and whose composition has not been modified since milking.

Q: *What does IPP denote in the context of environmental policies?*
A: Since then, new policy approaches on sustainable goods and services have been developed. These endeavours undertaken at all political levels have culminated in the Green Paper on Integrated Product Policy (1) (IPP). This document proposes a new strategy to strengthen and refocus product-related environmental policies and develop the market for greener products, which will also be one of the key innovative elements of the sixth environmental action programme - Environment 2010: “Our future, our choice”.

Reason. Reason questions ask for the reasons/motives/motivations for something happening. For example:

Q: *Why should the Regulation (EC) 1254 from 1999 be codified?*
A: (1) Commission Regulation (EC) No 562/2000 of 15 March 2000 laying down detailed rules for the application of Council Regulation (EC) No 1254/1999 as regards the buying-in of beef [3] has been substantially amended several times [4]. In the interests of clarity and rationality the said Regulation should be codified.

Q: *Why did a Commission expert conduct an inspection visit to Uruguay?*
A: A Commission expert has conducted an inspection visit to Uruguay to verify the conditions under which fishery products are produced, stored and dispatched to the Community.

Purpose. Purpose questions ask for the aim/goal/objective of something. For example:

Q: *What is the purpose of the Agreement of Luxembourg?*
A: RECALLING the object and purpose of the Agreement of Luxembourg to preserve the existing regime between the five Nordic States pursuant to the Convention on the Abolition of Passport Controls at Intra-Nordic borders signed in Copenhagen on 12 July 1957, establishing the Nordic Passport Union, once those of the Nordic States which are Members of the European Union take part in the regime on the abolition of checks on persons at internal borders set out in the Schengen agreements;"

Q: *What is the overall objective of the eco-label?*
A: The overall objective of the eco-label is to promote products which have the potential to reduce negative environmental impacts, as compared with the other products in the same product group, thus contributing to the efficient use of resources and a high level of environmental protection. In doing so it contributes to making consumption more sustainable, and to the policy objectives set out in the Community's sustainable development strategy (for example in the fields of climate change, resource efficiency and eco-toxicity), the sixth environmental action programme and the forthcoming White Paper on Integrated Product Policy Strategy.

Procedure. Procedure questions ask for a set of actions which is the official or accepted way of doing something. For example:

Q: *How are stable conditions in the natural rubber trade achieved?*
A: To achieve stable conditions in natural rubber trade through avoiding excessive natural rubber price fluctuations, which adversely affect the long-term interests of both producers and consumers, and stabilizing these prices without distorting long-term market trends, in the interests of producers and consumers;
Q: *What is the procedure for calling an extraordinary meeting?*
A: 2. Extraordinary meetings shall be convened by the Chairman if so requested by a delegation.

Q: *What is the common practice with shoots when packing them?*
A: (2) It is common practice in the sector to put white asparagus shoots into iced water before packing in order to avoid them becoming pink."

5 Test Set Preparation

Six hundred questions were initially formulated, manually verified against the document collection, translated into English and collected in a common xml format using a web interface specifically designed for this purpose. To avoid a bias towards a language, the 600 questions were developed by 6 different annotators originally in 6 different languages (100 each). All questions had at least one answer in the target corpus of that language.

In order to share them in a multilingual scenario, a second translation into all nine languages of the track was necessary. Native speakers from each language group with a good command of English were recruited and were asked to translate the questions

from English back into all the languages of the task. The final pool of 500 questions was selected by the track-coordinators out of the 600 produced, attempting to balance the question set according to the different question types (factoid, definition, reason, purpose and procedure). The need to select questions which had a supported answer in all the collections implied a great deal of extra work for the track coordinators, as a question collected in a language was not guaranteed to have an answer in all other collections.

During the creation of the 100 questions in a source language and their "mapping to English" the question creator was supposed not only to translate the questions into English, but also to look for the corresponding answer at least in the English corpus. After the selection of the final 500 questions, during their translation from English into the other source language, checking the availability of answers for all the questions in all the languages of the parallel corpus ensured that there is no NIL question, as in the previous QA@CLEF editions. The most frequent problematic situations were due to the misalignments between documents at the paragraph level:

- Entire paragraphs missing from one language, but existing in other(s); for example jrc31982D0886-ro contains only 25 paragraphs, but the English document contains 162 paragraphs, with the text containing an EC Convention, absent from the Romanian version.
- Different paragraph segmentation into different languages of the parallel corpus; for example the document jrc31985L0205-en contains one single paragraph (n="106") corresponding to 685 Romanian paragraphs (n="106_790"). From the point of view of our track, this means that one question having the answer in the (only one) English paragraph had to be removed, since the answer in Romanian is supposed to be found in exactly one paragraph.
- Missing information (parts of the text) in one paragraph; for example a question like "What should be understood by "living plants"?" had answer in English document jrc31968R0234-en paragraph number 8 "Whereas the production of live trees and other plants, bulbs, roots and the like, cut flowers and ornamental foliage (hereinafter where appropriate called 'live plants')". However, the corresponding Romanian paragraph number 9 does not include the list of the live plants.
- Contradictory information in corresponding paragraphs; for example the corresponding paragraphs that answers the question "How much does cotton increase in weight after treatment with formic acid?" indicate a loss of 3% in the Romanian version, whereas in English the loss is 4%.

6 Format

6.1 Test Set

Test sets for each source language took the form of a UTF-8 xml file containing the following:

```
source_lang    target_lang    q_id    q_string
```

where:

- source_lang is the source language
- target_lang is the target language
- q_id is the question number (4 digits – 0001 to 0500)
- q_string is the question (UTF-8 encoded) string

Here are four questions in a hypothetical EN-EN set:

```
<?xml version="1.0" encoding="UTF-8" ?>
<input>
  <q q_id="0001" source_lang="EN" target_lang="EN"> What should
  the driver of a Croatian heavy goods vehicle carry with him
  or her?</q>
  <q q_id="0002" source_lang="EN" target_lang="EN"> What will
  the Commission create under Regulation (EC) No 2422/2001 cre-
  ate? </q>
  <q q_id="0003" source_lang="EN" target_lang="EN"> What con-
  vention was done at Brussels on 15 December 1950? </q>
  <q q_id="0004" source_lang="EN" target_lang="EN"> What is an-
  other name for 'rights of transit'?</q>
</input>
```

6.2 Submission Format

A run submission file for the ResPubliQA task was also an xml file of the form:

```
q_id run_id answered passage-string p_id docid
```

where:

- q_id is the question number as given in the test set (of the form 0001 to 0500) Passages must be returned in the same ascending (increasing) order in which questions appear in the test set;
- run_id is the run ID an alphanumeric string which identifies the runs of each participant. It should be the concatenation of the following elements: the team ID (sequence of four lower case ASCII characters), the current year (09 stands for 2009), the number of the run (1 for the first one, or 2 for the second one), the task identifier (including both source and target languages, as in the test set);
- answered indicates if question has been answered or not. If the value for the attribute "answered" is NO, then the passage string will be ignored;
- passage_string is a text string; the entire paragraph which encloses the answer to the question;
- p_id is the number of the paragraph from which the passage_string has been extracted;
- docid is the ID of the document

i.e.

```
<?xml version="1.0" encoding="UTF-8" ?>
<output>
<a q_id="0001-0500" run_id="XXXX091XXXX" answered="YES|NO">
```

```
<passage_string p_id="11" docid "jrc31960D051-
en.xml">xyz</passage_string>
</a>
</output>
```

As it can be seen, systems were not required to answer all questions. See later for further discussion.

7 Evaluation

7.1 Responses

In this year's evaluation campaign, participants could consider questions and target collections in any language. Participants were allowed to submit just one response per question and up to two runs per task. Each question had to receive one of the following system responses:

1. A paragraph with the candidate answer. Paragraphs are marked and identified in the documents by the corresponding XML marks.
2. The string NOA to indicate that the system preferred not to answer the question.

Optionally, systems that preferred to leave some questions unanswered, could decide to submit also the candidate paragraph. If so, systems were evaluated for the responses they returned also in the cases in which they opted not to answer. This second option was used to additionally evaluate the validation performance.

One of the principles that inspired the evaluation exercise is that leaving a question unanswered has more value than giving a wrong answer. In this way, systems able to reduce the number of wrong answers, by deciding not to respond to some questions are rewarded by the evaluation measure.

However, a system choosing to leave some questions unanswered, returning NOA as a response, must ensure that only the portion of wrong answers is reduced, maintaining as high as possible the number of correct answers. Otherwise, the reduction in the number of correct answers is punished by the evaluation measure.

7.2 Assessments

Each run was manually judged by one human assessor for each language group, who considered if the paragraph was responsive or not. Answers were evaluated anonymously and simultaneously for the same question to ensure that the same criteria are being applied to all systems. This year, no second annotation was possible, so no data about the inter-annotator agreement are available.

One of the following judgements was given to each question-answer by human assessors during the evaluation:

- R : the question is answered correctly
- W: the question is answered incorrectly
- U : the question is unanswered

The evaluators were guided by the initial "gold" paragraph, which contained the answers. This "gold" paragraph was only a hint, since there were many cases when:

- correct answers did not exactly correspond to the "gold" paragraph, but the correct information was found in another paragraph of the same document as the "gold" one;
- correct answers corresponded to the "gold" paragraph, but were found in another JRC document;
- answers were evaluated as correct, even if the paragraphs returned contained more or less information than the "gold" paragraph;
- answers from different runs were evaluated as correct, even if they contained different but correct information; for example the question 44 (Which country wishes to export gastropods to the Community?) had Jamaica as the "gold" answer; but in the six runs evaluated, all the answers indicated Chile and Republic of Korea, which were also correct.

7.3 Evaluation Measure

The use of Machine Learning-based techniques able to decide if a candidate answer is finally acceptable or not was introduced by the Answer Validation Exercise[4] during the past campaigns. This is an important achievement, as an improvement in the accuracy of such decision-making process leads to more powerful QA architectures with new feedback loops. One of the goals of the ResPubliQA exercise is to effectively introduce these techniques in current QA systems.

For this reason, the unique measure considered in this evaluation campaign was the following:

$$c@1 = \frac{1}{n}(n_R + n_U \frac{n_R}{n})$$

where:

nR: is the number of correctly answered questions
n_U: number of unanswered questions
n: the total number of questions

Notice that this measure is parallel to the traditional accuracy used in past editions. The interpretation of the measure is the following:

1. A system that gives an answer to all the questions receives a score equal to the accuracy measure used in the previous QA@CLEF main task: in fact, since in this case $n_U = 0$ then $c@1 = n_R/n$;
2. The unanswered questions add value to c@1 only if they do not reduce much the accuracy (i.e. n_R/n) that the system would achieve responding to all questions. This can be thought as a hypothetical second chance in which the system would

[4] http://nlp.uned.es/clef-qa/ave

be able to replace some NoA answers by the correct ones. How many, the same proportion showed before (i.e. n_R/n).
3. A system that does not respond any question (i.e. returns only NOA as answer) receives a score equal to 0, as $n_R=0$ in both addends.

7.4 Tools and Infrastructure

This year, CELCT has developed a series of infrastructures to help the management of the ResPubliQA exercise. We had to deal with many processes and requirements:

- First of all the need to develop a proper and coherent tool for the management of the data produced during the campaign, to store it and to make it re-usable, as well as to facilitate the analysis and comparison of the results.
- Secondly, the necessity of assisting the different organizing groups in the various tasks of the data set creation and to facilitate the process of collection and translation of questions and their assessment.
- Finally, the possibility for the participants to directly access the data, submit their own runs (this also implied some syntax checks of the format), and later, get the detailed viewing of the results and statistics.

A series of automatic web interfaces were specifically designed for each of these purposes, with the aim of facilitating the data processing and, at the same time, showing the users only what is important for the task they had to accomplish. So, the main characteristics of these interfaces are the flexibility of the system specifically centred on the user's requirements.

While designing the interfaces for question collection and translation one of the first issues which was to be dealt with, was the fact of having many assessors, a big amount of data, and a long process. So tools must ensure an efficient and consistent management of the data, allowing:

1. Edition of the data already entered at any time.
2. Revision of the data by the users themselves.
3. Consistency propagation ensuring that modifications automatically re-model the output in which they are involved. For example, if a typo is corrected in the Translation Interface, the modification is automatically updated also in the Gold-Standard files, in the Test Set files and so on.
4. Statistics and evaluation measures are calculated and updated in real time.

8 Participants

11 groups participated with 28 runs. In addition, we evaluated 16 baseline runs (2 per language) based only in pure IR approach, for comparison purposes. All runs were monolingual except two runs Basque-English (EU-EN).

The most chosen language appeared to be English with 12 submitted runs, followed by Spanish with 6 submissions. No runs were submitted either in Bulgarian or Portuguese. Participants came above all from Europe, except two different groups from India. Table 1 shows the run distribution in the different languages.

Table 1. Tasks and corresponding numbers of submitted runs

		Target languages (corpus and answer)							
		BG	DE	EN	ES	FR	IT	PT	RO
Source languages (questions)	BG								
	DE		2						
	EN			10					
	ES				6				
	EU			2					
	FR					3			
	IT						1		
	PT								
	RO								4

The list of participating systems, teams and the reference to their reports are shown in Table 2.

Table 2. Systems and teams with the reference to their reports

System	Team	Reference
elix	ELHUYAR-IXA, SPAIN	Agirre et al., [1]
icia	RACAI, ROMANIA	Ion et al., [6]
iiit	Search & Info Extraction Lab, INDIA	Bharadwaj et al., [2]
iles	LIMSI-CNRS-2, FRANCE	Moriceau et al., [7]
isik	ISI-Kolkata, INDIA	-
loga	U.Koblenz-Landau, GERMAN	Gloeckner and Pelzer, [4]
mira	MIRACLE, SPAIN	Vicente-Díez et al., [15]
nlel	U. Politecnica Valencia, SPAIN	Correa et al., [3]
syna	Synapse Developpment, FRANCE	-
uaic	AI.I.Cuza U. of IASI, ROMANIA	Iftene et al., [5]
uned	UNED, SPAIN	Rodrigo et al., [13]

9 Results

9.1 IR Baselines

Since there were a parallel collection and one set of questions for all languages, the only variable that did not permit strict comparison between systems was the language itself. Running exactly the same IR system in all languages did not permit to fix this variable but at least we have some evidence about the starting difficulty in each language.

Two baseline runs per language, based on pure Information Retrieval, were prepared and assessed with two objectives:

1. to test how well can a pure Information Retrieval system perform on this task.
2. to compare the performance of more sophisticated QA technologies against a simple IR approach.

These baselines were produced in the following way:

1. Indexing the document collection at the paragraph level. Stopwords were deleted in all cases and the difference between the two runs is the application or not of stemming techniques.
2. Querying with the exact text of each question as a query.
3. Returning the paragraph retrieved in the first position of the ranking as the answer to the question.

The selection of an adequate retrieval model that fits the specific characteristic of the supplied data was a core part of the task. Applying an inadequate retrieval function would return a subset of paragraphs where the answer could not appear, and thus the subsequent techniques applied in order to detect the answer within the subset of candidates paragraphs would fail. For example, we found that simple models as the Vector Space Model or the default model of Lucene are not appropriate for this collection. For this reason, the baselines were produced using the Okapi-BM25 ranking function [11].

Using Okapi-BM25 the selection of the appropriate values for its parameters is crucial for a good retrieval. The parameters were fixed to:

1. b: 0.6. Those paragraphs with a length over the average obtain a slightly higher score.
2. k1: 0.1. The effect of term frequency over final score is minimised.

The same parameters in all runs for all languages were used. For more details about the preparation of these baselines see [10].

9.2 Results per Language

Tables 3-8 show systems performance divided by language. The content of the columns is as follows:

- **#R**: Number of questions answered correctly.
- **#W**: Number of questions answered wrongly.
- **#NoA**: Number of questions unanswered.
- **#NoA R**: Number of questions unanswered in which the candidate answer was Right. In this case, the system took the bad decision of leaving the question unanswered.
- **#NoA W**: Number of questions unanswered in which the candidate answer was Wrong. In this case, the system took a good decision leaving the question unanswered.
- **#NoA empty**: Number of questions unanswered in which no candidate answer was given. Since all questions had an answer, these cases were counted as if the candidate answer were wrong for accuracy calculation purpose.

- **c@1**: Official measure as it was explained in the previous section.
- **Accuracy**: The proportion of correct answers considering also the candidate answers of unanswered questions. That is:

$$accuracy = \frac{R + NoA\_R}{N}$$

where N is the number of questions (500).

Besides systems, there are three additional rows in each table:

- **Combination**: is the proportion of questions answered by at least one system or, in other words, the score of a hypothetical system doing the perfect combination of the runs.
- **Base091**: IR baseline as explained above, without stemming.
- **Base092**: IR baseline with stemming.

Table 3. Results for German

System	c@1	Accuracy	#R	#W	#NoA	#NoA R	#NoA W	#NoA empty
combination	0.56	0.56	278	222	0	0	0	0
loga091dede	0.44	0.4	186	221	93	16	68	9
loga092dede	0.44	0.4	187	230	83	12	62	9
base092dede	0.38	0.38	189	311	0	0	0	0
base091dede	0.35	0.35	174	326	0	0	0	0

Table 4. Results for English

System	c@1	Accuracy	#R	#W	#NoA	#NoA R	#NoA W	#NoA empty
combination	0.9	0.9	451	49	0	0	0	0
uned092enen	0.61	0.61	288	184	28	15	12	1
uned091enen	0.6	0.59	282	190	28	15	13	0
nlel091enen	0.58	0.57	287	211	2	0	0	2
uaic092enen	0.54	0.52	243	204	53	18	35	0
base092enen	0.53	0.53	263	236	1	1	0	0
base091enen	0.51	0.51	256	243	1	0	1	0
elix092enen	0.48	0.48	240	260	0	0	0	0
uaic091enen	0.44	0.42	200	253	47	11	36	0
elix091enen	0.42	0.42	211	289	0	0	0	0
syna091enen	0.28	0.28	141	359	0	0	0	0
isik091enen	0.25	0.25	126	374	0	0	0	0
iiit091enen	0.2	0.11	54	37	409	0	11	398
elix092euen	0.18	0.18	91	409	0	0	0	0
elix091euen	0.16	0.16	78	422	0	0	0	0

The system participating in the German task performed better than the baseline, showing a very good behaviour detecting the questions it could not answer. In 73% of unanswered questions (83% if we consider empty answers) the candidate answer was in fact incorrect. This shows the possibility of system improvement in a short time, adding further processing to the answering of questions predicted as unanswerable.

The first noticeable result in English is that 90% of questions received a correct answer by at least one system. However, this perfect combination is 50% higher than the best system result. This shows that the task is feasible but the systems still have room for improvement. Nevertheless, 0.6 of c@1 and accuracy is a result aligned with the best results obtained in other tasks of QA in the past campaigns of CLEF.

English results are indicative of the difference between c@1 and Accuracy values. The system uaic092 answered correctly 20 questions less than the baselines. However, this system was able to reduce the number of incorrect answers in a significant way, returning 32 incorrect answers less than the baselines. This behaviour is rewarded by c@1, producing a swap in the rankings (with respect to accuracy) between these two systems.

Another example is given by systems uaic091 and elix091, where the reduction of incorrect answers by uaic091 is significant in the case of with respect to elix091.

Something very interesting in the English runs is that the two best teams (see uned092enen, nlel091enen runs) produced paragraph rankings considering matching n-grams between question and paragraph [3]. This retrieval approach seems to be promising, since combined with paragraph validation filters it achieved the best score [13] in English.

These two approaches obtained the best score also in Spanish (uned091eses, nlel091eses). Additionally, [3] performed a second experiment (nlel092eses) that achieved the best result considering the whole parallel collection to obtain a list of answers in different languages (Spanish, English, Italian and French).

Table 5. Results for Spanish

System	c@1	Accuracy	#R	#W	#NoA	#NoA R	#NoA W	#NoA empty
combination	0.71	0.71	355	145	0	0	0	0
nlel092eses	0.47	0.44	218	248	34	0	0	34
uned091eses	0.41	0.42	195	275	30	13	17	0
uned092eses	0.41	0.41	195	277	28	12	16	0
base092eses	0.4	0.4	199	301	0	0	0	0
nlel091eses	0.35	0.35	173	322	5	0	0	5
base091eses	0.33	0.33	166	334	0	0	0	0
mira091eses	0.32	0.32	161	339	0	0	0	0
mira092eses	0.29	0.29	147	352	1	0	0	1

The experiment consisted in searching the questions in all languages, first selecting the paragraph with the highest similarity and then, returning the corresponding paragraph aligned in Spanish. This experiment obtained the best score in Spanish, opening the door to exploit the multilingual and parallel condition of the document collection.

In the case of French, baseline runs obtained the best results. Unexpectedly, Synapse (syna091frfr) usually obtaining the best scores in the news domain, did not perform well in this exercise. This proves that there are difficulties in moving from one domain into another.

Table 6. Results for French

System	c@1	Accuracy	#R	#W	#NoA	#NoA R	#NoA W	#NoA empty
combination	0.69	0.69	343	157	0	0	0	0
base092frfr	0.45	0.45	223	277	0	0	0	0
base091frfr	0.39	0.39	196	302	2	2	0	0
nlel091frfr	0.35	0.35	173	316	11	0	0	11
iles091frfr	0.28	0.28	138	362	0	0	0	0
syna091frfr	0.23	0.23	114	385	1	0	0	1

Table 7. Results for Italian

System	c@1	Accuracy	#R	#W	#NoA	#NoA R	#NoA W	#NoA empty
combination	0.61	0.61	307	193	0	0	0	0
nlel091itit	0.52	0.51	256	237	7	0	5	2
base092itit	0.42	0.42	212	288	0	0	0	0
base091itit	0.39	0.39	195	305	0	0	0	0

With respect to Italian (Table 7), the only participant obtained better results than the baselines.

Table 8. Results for Romanian

System	c@1	Accuracy	#R	#W	#NoA	#NoA R	#NoA W	#NoA empty
combination	0.76	0.76	381	119	0	0	0	0
icia092roro	0.68	0.52	260	84	156	0	0	156
icia091roro	0.58	0.47	237	156	107	0	0	107
UAIC092roro	0.47	0.47	236	264	0	0	0	0
UAIC091roro	0.45	0.45	227	273	0	0	0	0
base092roro	0.44	0.44	220	280	0	0	0	0
base091roro	0.37	0.37	185	315	0	0	0	0

The best system in Romanian [6] showed a very good performance compared to the rest of runs, as Table 8 shows. This is a system that uses a sophisticated similarity based model for paragraph ranking, question analysis, classification and regeneration of the question, classification of paragraphs and consideration of the EUROVOC terms associated to each document.

9.3 Comparison of Results Across Languages

Strict comparison between systems across languages is not possible without ignoring the language variable. However, this is the first time that systems working in different languages were evaluated with the same questions over the same document collection manually translated into different languages. So, extracting information about which approaches are more promising should be possible.

For this purpose, we considered both the systems participating in more than one language and the baseline IR runs for all languages.

Furthermore, the organization did not impose special restrictions to make use of a specific language or a combination of more languages. At the end, it can be said that the system that gave more correct answers and less incorrect ones is the best one, regardless of the language. However, the purpose is to compare approaches and follow the more promising one. Tables 9 and 10 mix all systems in all languages and rank them together in two dimensions, the value of c@1, and the target language.

Table 9. c@1 in participating systems according to the language

System	BG	DE	EN	ES	FR	IT	PT	RO
icia092								0.68
nlel092				0.47				
uned092			0.61	0.41				
uned091			0.6	0.41				
icia091								0.58
nlel091			0.58	0.35	0.35	0.52		
uaic092			0.54					0.47
loga091		0.44						
loga092		0.44						
base092	0.38	0.38	0.53	0.4	0.45	0.42	0.49	0.44
base091	0.38	0.35	0.51	0.33	0.39	0.39	0.46	0.37
elix092			0.48					
uaic091			0.44					0.45
elix091			0.42					
mira091				0.32				
mira092				0.29				
iles091					0.28			
syna091			0.28		0.23			
isik091			0.25					
iiit091			0.2					
elix092euen			0.18					
elix091euen			0.16					

In the first table (Table 9) systems are ordered by c@1 values. Reading column by column, systems are correctly ordered in each language, except some swaps with respect to the baseline IR suns. Systems icia092, uned and nlel seem to have the more powerful approaches.

Table 10. C@1/Best IR baseline

System	DE	EN	ES	FR	IT	RO
icia092						1.55
icia091						1.32
nlel092			1.18			
loga091	1.16					
loga092	1.16					
uned092		1.15	1.03			
uned091		1.13	1.03			
nlel091		1.09	0.88	0.78	1.24	
uaic092		1.02				1.07
elix092		0.91				
uaic091		0.83				1.02
mira091			0.80			
elix091		0.79				
mira092			0.73			
iles091				0.62		
syna091		0.53		0.51		
isik091		0.47				
iiit091		0.38				
elix092euen		0.34				
elix091euen		0.30				

Table 11. Number of questions answered by systems in different languages

Number of Languages	Questions Answered
0	6
1	20
2	45
3	52
4	55
5	76
6	76
7	96
8	74

In the next table (Table 10) we tried to partially fix the language variable, dividing c@1 values by the score of the best IR baseline system. Values over 1 indicate better performance than the baseline, and values under 1 indicate worse performance than the baseline.

Table 12. Methods used by participating systems

System name	Question Analyses			Retrieval Model	Linguistic Unit which is indexed		
	No Question Analysis	Manually done Patterns	Other		Words	Lemmas	Stems
SYNA		x		question category		x	
ICIA			MaxEnt question classification, automatic query generation using POS tagging and chunking	Boolean search engine	x	x	
ISIK	x			DFR	x		
NLEL	x			Clustered Keywords Positional Distance model			
UAIC		x			x	x	
MIRA		x		Vector			x
ILES		x				x	
IIIT		x	statistical method	Boolean model	x		x
UNED		x	Question classification	Okapi BM25			x
ELIX			Basque lemmatizer	Okapi BM25			x
LOGA		x	classification rules applied to question parse	Lucene, sentence segmentation. Also indexes contained answer types of a sentence		x	

In Table 10, the ranking of systems change, showing that also system loga proposes a promising approach, whereas nlel091 system appears more aligned with the baselines than loga. Of course, this evidence is affected by another variable that must be taken into account before making strong claims, i.e. the baseline itself, which perhaps is not the best approach for all languages (especially agglutinative languages such as German).

Table 11 shows the number of questions that have been correctly answered in only a certain number of languages. That is, for example 20 questions have been correctly answered in only one language (some of them in only a certain language, some of the rest in only another language, etc). This Table shows that the majority of questions have been answered by systems in many different languages. For example, 74 questions have been answered in all languages, whereas only 6 questions remained unanswered considering all languages. Notice that 99% of questions have been answered by at least one system in at least one language.

10 System Descriptions

Tables 12 and 13 summarise the characteristics of the participant systems. As can be seen, some systems did not analyse the questions at all. Among those that did, the most popular technique was the use of manually created query patterns (e.g. "Where is..." could indicate a location question). As regards retrieval models, two systems used Boolean methods while the rest mainly used Okapi or a VSM-type model.

Table 13 shows the type of processing techniques which were used on document fragments returned by the information retrieval components. As would be expected, Named Entity recognition and Numerical Expression recognition were widely used approaches.

Table 13. Methods used by systems for extracting answers

Answer Extraction – Further processing													
System name	**Chunking**	**n-grams**	**Named Entity Recognition**	**Temporal expressions**	**Numerical expressions**	**Dependency analysis**	**Functions (sub, obj,)**	**Syntactic transformations**	**Semantic parsing**	**Semantic role labeling**	**Logic representation**	**Theorem prover**	**None**
SYNA			x	x	x	x	x		x	x			
ICIA			x		x								
ISIK													x
NLEL													x
UAIC			x	x	x								
MIRA			x	x	x								
ILES	x		x		x	x	x	x					
IIIT	x		x		x								
UNED		x	x	x	x								
ELIX													
LOGA				x	x				x	x	x	x	

Table 14 shows the types of technique used for the answer validation component. Some systems did not have such a component, but for those that did, lexical similarity and syntactic similarity were the most widely used approaches.

11 Open Issues

Whereas in previous years, almost all responses were double-blind evaluated to check inter-evaluator agreement, this year it was not possible. A measure of the

inter-annotator agreement would have provided us an idea of the complexity and ambiguity of both questions and their supporting passages.

Moreover, this was the first year of using the JRC-Acquis collection which claims to be parallel in all languages. The supposed advantage of this was that all systems answer the same questions against the same document collections. Only the language of the questions and documents vary as otherwise the text is supposed to mean exactly the same. However, we found that in fact the texts are not parallel, being many passages left out or translated in a completely different way. The result was that many questions were not supported in all languages and could not therefore be used. This problem resulted in a huge amount of extra work for the organisers. Furthermore, the character of the document collection necessitated changes to the type of the questions. In most cases the questions became more verbose in order to deal with the vagueness and ambiguity of texts.

The idea of introducing new question types Reason, Purpose and Procedure was good in principle, but it did not seem to work as expected. Reason and Purpose questions resulted to be understood as more or less the same and the way in which these reasons and purposes are stated in the documents sometimes is meaningless. A typical type of reason is "to ensure smooth running of the EU" and a typical purpose is "to implement such and such a law". With respect to procedures there were also some non informative responses similar to the idea "the procedure to apply the law is to put it into practice".

Table 14. Technique used for the Answer Validation component

	Answer Validation							
System name	**No answer validation**	**Machine Learning is used to validate answers**	**Combined classifiers, Minimum Error Rate Training**	**Redundancies in the collection**	**Lexical similarity (term overlapping)**	**Sintactic similarity**	**Semantic similarity**	**Theorem proof or similar**
SYNA				x				
ICIA		x	x		x	x	x	
ISIK	x							
NLEL	x							
UAIC					x	x		
MIRA	x							
ILES				x	x	x		
IIIT					x	x		
UNED								
ELIX	x							
LOGA		x	x	x	x			x

Finally, the user model is still unclear, even after checking the kind of questions and answers that were feasible with the current setting: neither lawyers or ordinary people would not ask the kind of questions proposed in the exercise. Once more, the problem is to find the trade-off between research and a user centred development.

12 Conclusions

494 questions (99%) were answered by at least one system in at least one language; nevertheless the systems that gave more correct answers only answered 288. This shows that the task is feasible and systems still have room to improve and solve it in a short time.

One of the main issues is the retrieval model. Many systems must pay more attention to it since they performed worse than the baselines based on just IR. From this perspective, paragraph ranking approaches based on n-grams seems promising.

Some systems are able to reduce the number of incorrect answers maintaining a similar level in the number of correct answers, just leaving some questions unanswered. We expect this to be a first step towards the improvement of systems. This ability has been rewarded by the c@1 measure. Finally, moving to a new domain has raised new questions and challenges for both organizers and participants.

Acknowledgments

This work has been partially supported by the TrebleCLEF Coordination Action, within FP7 of the European Commission, Theme ICT-1-4-1 Digital Libraries and Technology Enhanced Learning (Contract 215231), the Education Council of the Regional Government of Madrid and the European Social Fund.

Special thanks are due to: Luís Costa, Luís Cabral, Diana Santos and two German students (Anna Kampchen and Julia Kramme) for taking care of the translations of the questions and the evaluation of the submitted runs for the Portuguese and German languages respectively.

Special thanks are also due to Cosmina Croitoru, a bright Romanian student whose help in the answers evaluation permitted to detect about 5 evaluation errors and some unevaluated answers in the RO-RO runs.

Our appreciation also to the advisory board: Donna Harman (NIST, USA), Maarten de Rijke (University of Amsterdam, The Netherlands), Dominique Laurent (Synapse Développement, France)

References

1. Agirre, E., Ansa, O., Arregi, X., de Lacalle, M.L., Otegi, A., Saralegi, X., Zaragoza, H.: Elhuyar-IXA: Semantic Relatedness and Cross-lingual Passage Retrieval. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece.A, September 30 -October 2 (2009)
2. Bharadwaj, R., Ganesh, S., Varma, V.: A Naïve Approach for Monolingual Question Answering. In: Working Notes for the CLEF 2009, Workshop, Corfu, Greece, September 30 - October 2, (2009)

3. Correa, S., Buscaldi, D., Rosso, P.: NLEL-MAAT at CLEF-ResPubliQA. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30 -October 2 (2009)
4. Gloeckner, I., Pelzer, B.: The LogAnswer Project at CLEF. In: Working Notes for the CLEF 2009, Workshop, Corfu, Greece, September 30 - October 2 (2009)
5. Iftene, A., Trandabăţ1, D., Pistol, I., Moruz1, A.-M., Husarciuc1, M., Sterpu, M., Turliuc, C.: Question Answering on English and Romanian Languages. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)
6. Ion, R., Ştefănescu, D., Ceauşu, A., Tufiş, D., Irimia, E., Barbu-Mititelu, V.: A Trainable Multi-factored QA System. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)
7. Moriceau, V., Tannier, X.: FIDJI in ResPubliQA 2009. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)
8. Peñas, A., Rodrigo, Á., Verdejo, F.: Overview of the Answer Validation Exercise 2007. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 237–248. Springer, Heidelberg (2008)
9. Peñas, A., Rodrigo, Á., Sama, V., Verdejo, F.: Overview of the Answer Validation Exercise 2006. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 257–264. Springer, Heidelberg (2007)
10. Pérez, J., Garrido, G., Rodrigo, Á., Araujo, L., Peñas, A.: Information Retrieval Baselines for the ResPubliQA Task. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)
11. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: SIGIR 1994: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 232–241 (1994)
12. Rodrigo, Á., Peñas, A., Verdejo, F.: Overview of the Answer Validation Exercise 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 296–313. Springer, Heidelberg (2009)
13. Rodrigo, Á., Pérez, J., Peñas, A., Garrido, G., Araujo, L.: Approaching Question Answering by means of Paragraph Validation. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)
14. Tomlinson, S., Oard, D.W., Baron, J.R., Thompson, P.: Overview of the TREC 2007 Legal Track. In: Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, November 5-9 (2007)
15. Vicente-Díez, M.T., de Pablo-Sánchez, C., Martínez, P., Schneider, J.M., Salazar, M.G.: Are Passages Enough? The MIRACLE Team Participation at QA@CLEF2009. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)

Overview of QAST 2009

Jordi Turmo[1], Pere R. Comas[1], Sophie Rosset[2], Olivier Galibert[2], Nicolas Moreau[3], Djamel Mostefa[3], Paolo Rosso[4] and Davide Buscaldi[4]

[1] TALP Research Centre (UPC). Barcelona. Spain
turmo@lsi.upc.edu, pcomas@lsi.upc.edu
[2] LIMSI. Paris. France
rosset@limsi.fr, olivier.galibert@limsi.fr
[3] ELDA/ELRA. Paris. France
moreau@elda.org, mostefa@elda.org
[4] NLE Lab. - ELiRF Research Group (UPV). Spain
prosso@dsic.upv.es, dbuscaldi@dsic.upv.es

Abstract. This paper describes the experience of QAST 2009, the third time a pilot track of CLEF has been held aiming to evaluate the task of Question Answering in Speech Transcripts. Four sites submitted results for at least one of the three scenarios (European Parliament debates in English and Spanish and broadcast news in French). In order to assess the impact of potential errors of automatic speech recognition (ASR), for each task manual transcripts and three different ASR outputs were provided. In addition an original method of question creation was tried in order to get spontaneous oral questions resulting in two sets of questions (spoken and written). Each participant who had chosen a task was asked to submit a run for each condition. The QAST 2009 evaluation framework is described, along with descriptions of the three scenarios and their associated data, the system submissions for this pilot track and the official evaluation results.

1 Introduction

Question Answering (QA) technology aims at providing answers to natural language questions. Current QA technology is focused mainly on the mining of written text sources for extracting the answer to written questions from both open-domain and restricted-domain document collections [7,3]. However, most human interaction occurs through speech, e.g. meetings, seminars, lectures, or telephone conversations. All these scenarios provide large amounts of information that could be mined by QA systems. As a consequence, the exploitation of speech sources brings QA a step closer to many real world applications in which spontaneous oral questions or written questions can be involved. The QAST 2009 track aims at investigating the problem of answering spontaneous oral questions and written questions using audio documents.

Current text-based QA systems tend to use technologies that require text written in accordance with standard norms for written grammar. The syntax of speech is quite different than that of written language, with more local but

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 197–211, 2010.

less constrained relations between phrases, and punctuation, which gives boundary cues in written language, is typically absent. Speech also contains disfluencies, repetitions, restarts and corrections. Moreover, any practical application of search in speech requires the transcriptions to be produced automatically, and the Automatic Speech Recognizers (ASR) introduce a number of errors. Therefore current techniques for text-based QA need substantial adaptation in order to access the information contained in audio documents, and probably to analyse oral questions. Preliminary research on QA in speech transcriptions was addressed in QAST 2007 and QAST 2008, pilot evaluation tracks at CLEF in which systems attempted to provide answers to written factual and definitional written questions by mining speech transcripts of different scenarios [5,6].

This paper provides an overview of the third QAST pilot evaluation. Section 2 describes the principles of this evaluation track. Sections 3 and 4 present the evaluation framework and the systems that participated, respectively. Section 5 reports and discusses the achieved results, followed by some conclusions in Section 6.

2 The QAST 2009 Task

The aim of this third year of QAST is to provide a framework in which QA systems can be evaluated in a real scenario, where the answers of both spontaneous oral questions and written questions have to be extracted from speech transcriptions, these transcriptions being manually and automatically generated. There are five main objectives to this evaluation:

- Motivating and driving the design of novel and robust QA architectures for speech transcripts;
- Measuring the loss due to the inaccuracies in state-of-the-art ASR technology;
- Measuring this loss at different ASR performance levels given by the ASR word error rate;
- Measuring the loss when dealing with spontaneous oral questions;
- Motivating the development of monolingual QA systems for languages other than English.

In the 2009 evaluation, as in the 2008 evaluation, an answer is structured as a simple [answer string, document id] pair where the answer string contains nothing more than the full and exact answer, and the document id is the unique identifier of the document supporting the answer. For the tasks on automatic speech transcripts, the answer string consisted of the <start-time> and the <end-time> giving the position of the answer in the signal.

Figure 1 illustrates this point. Given the manually transcribed spontaneous oral question *When did the bombing of Fallujah eee took take place?* corresponding to the written question *When did the bombing of Fallujah take place?*, the figure compares the expected answer in a manual transcript (the text *a week ago*) and in an automatic transcript (the time segment *1081.588 1082.178*). Note that

Fallujah was wrongly recognized as *for the Chair* by the ASR. A system can provide up to 5 ranked answers per question.

Spontaneous oral question: ***When did the bombing of Fallujah eee took take place?***
Written question: ***When did the bombing of Fallujah take place?***

Manual transcript: *(%hesitation) a week ago President the American (%hesitation) occupation forces (%hesitation) m() m() m() marched into Fallujah and they (%hesitation) bombarded (%hesitation) m() murdered and have been persecuting everyone in the city .*
Answer: *a week ago*

Extracted portion of an **automatic transcript (CTM file format):**
(...)
20041115_1705_1735_EN_SAT 1 **1081.588** 0.050 a 0.9595
20041115_1705_1735_EN_SAT 1 1081.638 0.190 week 0.9744
20041115_1705_1735_EN_SAT 1 **1081.828** 0.350 ago 0.9743
20041115_1705_1735_EN_SAT 1 1082.338 0.630 President 0.9576
20041115_1705_1735_EN_SAT 1 1083.648 0.310 the 0.9732
20041115_1705_1735_EN_SAT 1 1084.008 0.710 American 0.9739
20041115_1705_1735_EN_SAT 1 1085.078 0.450 occupation 0.9739
20041115_1705_1735_EN_SAT 1 1085.528 0.640 forces 0.9741
20041115_1705_1735_EN_SAT 1 1086.858 1.730 and 0.9742
20041115_1705_1735_EN_SAT 1 1089.098 0.170 we 0.6274
20041115_1705_1735_EN_SAT 1 1089.308 0.480 must 0.9571
20041115_1705_1735_EN_SAT 1 1089.948 0.300 into 0.9284
20041115_1705_1735_EN_SAT 1 1090.368 0.130 for 0.3609
20041115_1705_1735_EN_SAT 1 1090.498 0.130 the 0.3609
20041115_1705_1735_EN_SAT 1 1090.698 0.240 Chair 0.2233
20041115_1705_1735_EN_SAT 1 1091.678 0.600 and 0.9755
20041115_1705_1735_EN_SAT 1 1092.798 0.400 they 0.9686
20041115_1705_1735_EN_SAT 1 1093.598 0.530 bombarded 0.8314
(...)
Answer: 1081.588 1081.828

Fig. 1. Example query and response from manual (top) and automatic (bottom) transcripts

A total of six tasks were defined for this third edition of QAST covering three scenarios: English questions related to European Parliament sessions in English (T1a and T1b), Spanish questions related to European Parliament sessions in Spanish (T2a and T2b) and French questions related to French Broadcast News (T3a and T3b). The complete set of tasks is:

– T1a: QA of English written questions in the manual and automatic transcriptions of European Parliament Plenary sessions in English (EPPS English corpus).

- T1b: QA of manual transcriptions of English spontaneous oral questions in the manual and automatic transcriptions of European Parliament Plenary sessions in English (EPPS English corpus).
- T2a: QA of Spanish written questions in the manual and automatic transcriptions of European Parliament Plenary sessions in Spanish (EPPS Spanish corpus).
- T2b: QA of manual transcriptions of Spanish spontaneous oral questions in the manual and automatic transcriptions of European Parliament Plenary sessions in Spanish (EPPS Spanish corpus).
- T3a: QA of French written questions in manual and automatic transcriptions of broadcast news for French (ESTER corpus).
- T3b: QA of manual transcriptions of French spontaneous oral questions in manual and automatic transcriptions of broadcast news for French (ESTER corpus)

3 Evaluation Protocol

3.1 Data Collections

The QAST 2009 data is derived from three different resources, each one corresponding to a different language (English, Spanish and French):

- English parliament (EPPS EN): The **TC-STAR05 EPPS English corpus** [4] contains 3 hours of recordings in English corresponding to 6 sessions of the European Parliament. The data was used to evaluated speech recognizers in the TC-STAR project. There are 3 different automatic speech recognition outputs with different word error rates (10.6%, 14% and 24.1%). The manual transcriptions were done by ELDA.
- Spanish parliament (EPPS ES): The **TC-STAR05 EPPS Spanish corpus** [4] is comprised of three hours of recordings in Spanish corresponding to 6 sessions of the European Parliament. The data was used to evaluate Spanish ASR systems developed in the TC-STAR project. There are 3 different automatic speech recognition outputs with different word error rates (11.5%, 12.7% and 13.7%). The manual transcriptions were done by ELDA.
- French broadcast news (French BN): The test portion of the **ESTER corpus** [2] contains 10 hours of broadcast news recordings in French, comprising 18 shows from different sources (France Inter, Radio France International, Radio Classique, France Culture, Radio Television du Maroc). There are 3 different automatic speech recognition outputs with different error rates (11.0%, 23.9% and 35.4%). The manual transcriptions were produced by ELDA.

These three collections are the same than the ones used last year for the QAST 2008 evaluation campaign.

European Parliament and Broadcast News data are usually referred to as *prepared speech*. Although they typically have few interruptions and turn-taking

problems when compared to actual *spontaneous speech*, many of the characteristics of spoken language are still present (hesitations, breath noises, speech errors, false starts, mispronunciations and corrections).

3.2 Questions and Answer Types

For each of the three languages, two sets of manually transcribed spontaneous oral questions and their respective written questions have been created and provided to the participants, the first for development purposes and the second for the evaluation:

- Development sets (released on the 25th of March 2009):
  - EPPS EN: 50 transcribed questions and their respective written questions.
  - EPPS ES: 50 transcribed questions and their respective written questions.
  - French BN: 50 transcribed questions and their respective written questions.
- Evaluation sets (released on the 1st of June 2009):
  - EPPS EN: 100 transcribed questions and their respective written questions.
  - EPPS ES: 100 transcribed questions and their respective written questions.
  - French BN: 100 transcribed questions and their respective written questions.

For each language, both the development and evaluation sets were created from the whole document collection (i.e. the 6 European Parliament sessions for English and Spanish, and the 18 Broadcast News shows for French). In other words, there was no collection split between a development data set and an evaluation data set as was done last year.

As for last year, two types of questions were considered: factual questions and definitional ones. The expected answer to a factual question is a *named entity*. There were 6 types of factual question this year, each corresponding to a particular category of named entities[1]:

- **Person:** names of humans, real and fictional, fictional or real non-human individuals.
  Examples: *Mirjam Killer*, *John*, *Jesus*, etc.
- **Organisation:** names of business, multinational organizations, political parties, religious groups, etc.
  Ex: *CIA*, *IBM*, but also named entities like *Washington* when they display the characteristics of an organisation.

[1] Although neither measures not time expressions are real named entities, they are usually considered as so in the state of the art.

- **Location:** geographical, political or astronomical entities.
  Ex: *California*, *South of California*, *Earth*, etc.
- **Time:** a date or a specific moment in time, absolute and relative time expressions.
  Ex: *March 28th*, *last week*, *at four o'clock in the morning*, etc. Deictic expressions have been considered given that they are commonly used as references for temporal values.
- **Measure:** measures of length, width or weight, etc. Generally, a quantity and a unit of measurement.
  Ex: *five kilometers*, *20 Hertz*, etc. But also ages, period of time, etc.

This is less than the 10 categories used for the 2007 and 2008 evaluations. Some categories have not been considered this year because no occurence were found in the collected set of sponteaneous questions (*Color*, *Shape*, *Language*, *System*, *Material*).

The definition questions are questions such as *What is the CDU?* and the answer can be anything. In this example, the answer would be *political group*. This year, the definition questions are subdivided into three types:

- **Person:** question about someone.
  Q: *Who is George Bush?*
  R: *The President of the United States of America.*
- **Organisation:** question about an organisation.
  Q: *What is Cortes?*
  R: *Parliament of Spain.*
- **Other:** questions about technology, natural phenomena, etc.
  Q: *What is the name of the system created by AT&T?*
  R: *The How can I help you system.*

For each language a number of 'NIL' questions (i.e., questions having no answer in the document collection) have been selected. The distribution of the different types of questions across the three collections is shown in Table 3.2.

Table 1. Distribution of question types per task: T1 (EPPS EN), T2 (EPPS ES), T3 (French BN)

Type	Factual	Definition	NIL
T1 (English)	75%	25%	18%
T2 (Spanish)	55%	45%	23%
T3 (French)	68%	32%	21%

The question sets are formatted as plain text files, with one question per line (see the QAST 2008 Guidelines[2]). The procedure to generate the questions is described in the following section.

[2] http://www.lsi.upc.edu/~qast:News

Question generation. A novel feature in QAST 2009 was the introduction of spontaneous oral questions. The main issue in the generation of this kind of questions was how to obtain spontaneity. The solution adopted was to set up the following procedure for question generation:

1. Passage generation: a set of passages was randomly extracted from the document collection. A single passage was composed by the complete sentences included in a text window of 720 characters.
2. Question generation: human question generators were randomly assigned a number of passages (varying from 2 to 4). They had to read each passage and then to formulate one or more questions based on the passage they just read about information not present in it.
3. Question transcription: precise transcriptions of the oral spontaneous questions were made, including hesitations, etc.
   Ex: *(%hesitation) What (%hesitation) house is the pres() the president elect being elected to?*
4. Question filtering: some questions were filtered out from the set of generated questions because their answer types were not allowed (causal and manner questions) or because they did not have answer in the document collection. The resulting questions were usable questions.
5. Written question generation: the usable questions were re-written by removing speech disfluencies, correcting the syntax and simplifying the sentence when necessary.
   Ex: *What house does the president run?*
6. Question selection: the final set of development questions and test questions were selected by ELDA from the usable questions.

The allowed question types were the following:

- *definition*: person, organisation, object and other.
- *factoid*: person, location, organisation, time (includes date), measure and language.

However, the types "language" for factual questions and "object" for definition questions did not occur among the generated questions.

A preliminary evaluation of the generated questions was carried out in order to determine how many usable questions could be produced by a human reader. The results of this evaluation show that the percentage of usable questions produced by the questions generator was between 47% and 58% of the total questions produced, depending on the speakers knowledge of the task guidelines. These figures show that the produced questions were more than the number of questions actually presented to participants in QAST 2009. Most unusable questions were due to the fact that human question generators *forgot* the guidelines many times while asking their questions. Table 2 shows the number of questions recorded, the resulting usable questions and the average of the length in words per question for each language.

Table 2. Details of the questions generated for each language

	#speaker	#questions recorded	#usable questions	avg. #words
English	12	1096	616	9.1
French	7	485	335	7.7
Spanish	11	403	313	7.1

3.3 Human Judgment

As in 2008, the answer files submitted by participants have been manually judged by native speaking assessors, who considered the correctness and exactness of the returned answers. They also checked that the document labeled with the returned document ID supports the given answer. One assessor evaluated the results, and another assessor manually checked each judgment of the first one. Any doubts about an answer was solved through various discussions. The assessors used the QASTLE[3] evaluation tool developed in Perl (at ELDA) to evaluate the systems' results. A simple window-based interface permits easy, simultaneous access to the question, the answer and the document associated with the answer.

After each judgment the submission files were modified by the interface, adding a new element in the first column: the answer's evaluation (or judgment). The four possible judgments (also used at TREC [7]) correspond to a number ranging between 0 and 3:

- 0 correct: the answer-string consists of the relevant information (exact answer), and the answer is supported by the returned document.
- 1 incorrect: the answer-string does not contain a correct answer.
- 2 inexact: the answer-string contains a correct answer and the docid supports it, but the string has bits of the answer missing or contains additional texts (longer than it should be).
- 3 unsupported: the answer-string contains a correct answer, but is not supported by the docid.

3.4 Measures

The two following metrics (also used in CLEF) were used in the QAST evaluation:

1. Mean Reciprocal Rank (MRR): This measures how well the right answer is ranked in the list of 5 possible answers.
2. Accuracy: The fraction of correct answers ranked in the first position in the list of 5 possible answers.

[3] http://www.elda.org/qastle/

4 Submitted Runs

A total of four groups from four different countries submitted results for one or more of the proposed QAST 2009 tasks. Due to various reasons (technical, financial, etc.), eight other groups registered, but were not be able to submit any results.

The four participating groups were:

- INAOE, Instituto Nacional de Astrofísica, Optica y Electríca, Mexico;
- LIMSI, Laboratoire d'Informatique et de Mécanique des Sciences de l'Ingénieur, France;
- TOK, Tokyo Institute of Technology, Japan;
- UPC, Universitat Politècnica de Catalunya, Spain.

All groups participated to task T1 (EPPS EN), UPC and LIMSI participated to task T2 (EPPS ES) and only LIMSI dealt with task T3 (French BN). Each participant could submit up to 48 submissions (2 runs per task and transcription). In order to allow comparisons on the performance of the systems when using different WER levels in the transcriptions, it was mandatory for each task to submit results for all the data: the manual transcriptions and the three ASR outputs (automatic transcriptions).

Table 3 shows the number of submitted runs per participant and task. The number of submissions ranged from 8 to 32. The characteristics of the systems used in the submissions are summarized in Table 5. More detailed information on the systems can be found in QAST 2009 working notes (http://www.clef-campaign.org/2009/working_notes/CLEF2009WN-Contents.html). A total of 86 submissions were evaluated with the distribution across tasks shown in the bottom row of the table.

Table 3. Submitted runs per participant and task

Participant	T1a	T1b	T2a	T2b	T3a	T3b
INAOE	8	8	-	-	-	-
LIMSI	5	5	5	5	5	5
TOK	4	4	-	-	-	-
UPC	8	8	8	8	-	-
Total	25	25	13	13	5	5

5 Results

The results for the three tasks in manual transcribed data are presented in Tables 5 to 7, according to the question types (factual, definitional and all questions).

The results for the three tasks in automatically transcribed data are presented in Tables 8 to 10, according to the question types (factual, definitional and all questions).

Table 4. Characteristics of the systems that participated in QAST 2009

System	Enrichment	Question classification	Doc./Passage Retrieval	Factual Answer Extraction	Def. Answer Extraction	NERC
INAOE1	words and NEs	hand-crafted rules	Indri	passage selection based on NEs of the question type	-	regular expressions
INAOE2	same plus phonetics					
LIMSI1	words, lemmas, morphologic derivations, synonymic relations and extended NEs	hand-crafted rules	passage ranking based on search descriptors	ranking based on distance and redundancy	specific index for known acronyms	hand-crafted rules with statistical POS
LIMSI2				ranking based on bayesian modelling		
TOK1	words and word classes derived from training data - question-answer pairs	-	sentence ranking based on statistical models	ranking based on analogy between input question and question in the training data	-	-
UPC1	words, NEs lemmas and POS	perceptrons	passage ranking through iterative query relaxation	ranking based on keyword distance and density	-	hand-crafted rules, gazeetters and perceptrons
UPC2	same plus phonetics		addition of approximated phonetic matching			

Table 5. Results for task T1, English EPPS, manual transcripts (75 factual questions and 25 definitional ones)

System	Questions	Factual			Definitional			All	
		#Correct	MRR	Acc	#Correct	MRR	Acc	MRR	Acc
INAOE1	Written	44	0.38	26.7%	10	0.31	28.0%	0.36	27%
	Spoken	28	0.27	21.3%	7	0.26	24.0%	0.27	22%
INAOE2	Written	42	0.38	28.0%	9	0.30	28.0%	**0.36**	**28%**
	Spoken	38	0.35	25.3%	9	0.30	28.0%	0.34	26%
LIMSI1	Written	42	0.39	29.3%	11	0.28	20.0%	0.36	27%
	Spoken	39	0.36	25.3%	10	0.24	16%	0.33	23%
LIMSI2	Written	32	0.31	22.7%	13	0.36	24.0%	0.32	23%
	Spoken	30	0.26	18.7%	11	0.30	20.0%	0.27	19%
TOK1	Written	11	0.10	6.7%	3	0.03	0.0%	0.08	5%
	Spoken	11	0.08	4.0%	3	0.03	0.0%	0.06	3%
UPC1	Written	32	0.27	18.7%	8	0.29	28.0%	0.28	21%
	Spoken	19	0.15	9.3%	2	0.05	4.0%	0.12	8%
UPC2	Written	35	0.31	22.7%	8	0.29	28.0%	0.31	24%
	Spoken	18	0.15	9.3%	2	0.05	4.0%	0.12	8%

Table 6. Results for task T2, Spanish EPPS, manual transcripts (44 factual questions and 56 definitional ones)

System	Questions	Factual			Definitional			All	
		#Correct	MRR	Acc	#Correct	MRR	Acc	MRR	Acc
LIMSI1	Written	32	0.56	45.5%	29	0.36	28.6%	**0.45**	**36.0%**
	Spoken	32	0.56	45.5%	30	0.37	28.6%	0.45	36.0%
LIMSI2	Written	26	0.41	29.5%	23	0.28	19.6%	0.34	24.0%
	Spoken	26	0.41	29.5%	23	0.28	19.6%	0.34	24.0%
UPC1	Written	16	0.24	15.9%	10	0.16	14.3%	0.20	15.0%
	Spoken	20	0.34	27.3%	9	0.13	10.7%	0.22	18.0%
UPC2	Written	20	0.29	18.2%	10	0.14	10.7%	0.20	14.0%
	Spoken	20	0.33	27.3%	9	0.13	8.9%	0.22	17.0%

Table 7. Results for task T3, French Broadcast News, manual transcripts (68 factual questions and 32 definitional ones)

System	Questions	Factual			Definitional			All	
		#Correct	MRR	Acc	#Correct	MRR	Acc	MRR	Acc
LIMSI1	Written	38	0.35	23.5%	22	0.47	37.5%	**0.39**	**28.0%**
	Spoken	39	0.36	23.5%	20	0.46	37.5%	0.39	28.0%
LIMSI2	Written	38	0.34	22.1%	22	0.47	37.5%	0.38	27.0%
	Spoken	39	0.36	23.5%	20	0.46	37.5%	0.39	28.0%

7 systems participated in the T1 (English) task on manual transcripts and 6 on automatic transcripts.

On manual transcripts, the accuracy ranged from 28% to 5% (for written questions) and from 26% to 3% (for spoken questions).

For five of the systems, we observe a relatively small difference between written and spoken questions (from 2% to 5% loss going from written questions to spoken questions). The other two systems encountered a significant loss (13% and 16% of difference between written and spoken questions).

There were three approaches for QA on automatic speech transcripts used by the systems. The LIMSI and UPC on all ASRs and INAOE on ASR_A and ASR_B took the ASR output at the only available information. INAOE on ASR_C used information extracted from all the ASR outputs, keeping ASR_C as primary. This approach could represent an application where multiple ASR outputs from different systems are available. Combining outputs from varied systems is a standard method in speech recognition to obtain a lower word error rate [1], it is interesting to see if the same kind of method can be used at a more semantic level. The TOK system on the other hand used sentence segmentation information from the manual transcripts and applied it to the automatic transcripts. While such a segmentation information is not available in the transcriptions given, ASR systems do generate an acoustically motivated segmentation as a step of their processing. The TOK approach could then be

Table 8. Results for task T1, English EPPS, automatic transcripts (75 factual questions and 25 definitional ones)

ASR	System	Questions	Factual			Definitional			All	
			#Correct	MRR	Acc	#Correct	MRR	Acc	MRR	Acc
ASR_A 10.6%	INAOE1	Written	35	0.32	24.0%	6	0.21	20.0%	0.30	23.0%
		Spoken	34	0.33	25.3%	6	0.21	20.0%	0.30	24.0%
	INAOE2	Written	35	0.32	22.7%	7	0.22	20.0%	0.29	22.0%
		Spoken	34	0.32	24.0%	7	0.22	20.0%	0.29	23.0%
	LIMSI1	Written	32	0.34	28.0%	10	0.25	20.0%	**0.31**	**26.0%**
		Spoken	30	0.31	25.3%	11	0.29	24.0%	0.30	25.0%
	TOK1	Written	13	0.08	4.0%	3	0.04	0.0%	0.07	3.0%
		Spoken	12	0.07	2.7%	4	0.08	4.0%	0.07	3.0%
	UPC1	Written	29	0.27	18.7%	7	0.26	24.0%	0.27	20.0%
		Spoken	11	0.08	5.3%	2	0.06	4.0%	0.08	5.0%
	UPC2	Written	30	0.26	18.7%	6	0.24	24.0%	0.26	20.0%
		Spoken	12	0.09	5.3%	1	0.04	4.0%	0.08	5.0%
ASR_B 14.0%	INAOE1	Written	23	0.22	16.0%	6	0.21	20.0%	0.22	17.0%
		Spoken	23	0.21	13.3%	7	0.25	24.0%	0.22	16.0%
	INAOE2	Written	24	0.22	16.0%	6	0.21	20.0%	0.22	17.0%
		Spoken	24	0.21	13.3%	7	0.25	24.0%	0.22	16.0%
	LIMSI1	Written	24	0.27	22.7%	8	0.20	16.0%	**0.25**	**21.0%**
		Spoken	24	0.26	21.3%	9	0.24	20.0%	0.25	21.0%
	TOK1	Written	9	0.06	4.0%	3	0.03	0.0%	0.06	3.0%
		Spoken	10	0.06	2.7%	3	0.06	4.0%	0.06	3.0%
	UPC1	Written	26	0.24	17.3%	7	0.26	24.0%	0.24	19.0%
		Spoken	11	0.08	4.0%	2	0.06	4.0%	0.08	4.0%
	UPC2	Written	29	0.26	20.0%	7	0.25	24.0%	0.26	21.0%
		Spoken	12	0.08	4.0%	2	0.05	4.0%	0.07	4.0%
ASR_C 24.1%	INAOE1	Written	29	0.31	26.7%	5	0.20	20.0%	**0.28**	**25.0%**
		Spoken	28	0.30	26.7%	5	0.20	20.0%	0.28	25.0%
	INAOE2	Written	29	0.30	25.3%	6	0.21	20.0%	0.28	24.0%
		Spoken	28	0.29	24.0%	6	0.21	20.0%	0.27	23.0%
	LIMSI1	Written	23	0.26	24.0%	8	0.19	12.0%	0.24	21.0%
		Spoken	24	0.24	21.3%	9	0.23	16.0%	0.24	20.0%
	TOK1	Written	17	0.12	5.3%	5	0.08	4.0%	0.11	5.0%
		Spoken	19	0.11	4.0%	5	0.12	8.0%	0.11	5.0%
	UPC1	Written	22	0.21	16.0%	6	0.24	24.0%	0.22	18.0%
		Spoken	10	0.08	5.3%	1	0.04	4.0%	0.07	5.0%
	UPC2	Written	26	0.24	17.3%	6	0.24	24.0%	0.24	19.0%
		Spoken	11	0.08	4.0%	1	0.04	4.0%	0.07	4.0%

considered as using an optimistic approximation of this automatically generated segmentation information. In any case, comparing systems and estimating the impact of WER can only be done on "pure" systems (LIMSI and UPC on all ASRs and INAOE on ASR_A and ASR_B).

On the ASR transcripts for the pure systems, the accuracy ranged for the best ASR (10.6% of WER) from 26% (written questions) to 5% (spoken questions).

Table 9. Results for task T2, Spanish EPPS, automatic transcripts (44 factual questions and 56 definitional ones)

ASR	System	Questions	Factual			Definitional			All	
			#Correct	MRR	Acc	#Correct	MRR	Acc	MRR	Acc
ASR_A 11.5%	LIMSI1	Written	20	0.37	31.8%	22	0.29	23.2%	**0.32**	**27.0%**
		Spoken	20	0.37	31.8%	21	0.27	21.4%	0.31	26.0%
	UPC1	Written	8	0.15	13.6%	2	0.01	0.0%	0.07	6.0%
		Spoken	6	0.14	13.6%	2	0.01	0.0%	0.07	6.0%
	UPC2	Written	12	0.20	18.2%	3	0.02	0.0%	0.10	8.0%
		Spoken	12	0.24	22.7%	3	0.03	1.8%	0.12	11.0%
ASR_B 12.7%	LIMSI1	Written	18	0.32	27.3%	19	0.26	23.2%	**0.29**	**25.0%**
		Spoken	18	0.32	27.3%	19	0.26	23.2%	0.29	25.0%
	UPC1	Written	12	0.18	13.6%	2	0.04	3.6%	0.10	8.0%
		Spoken	12	0.20	15.9%	1	0.02	1.8%	0.10	8.0%
	UPC2	Written	13	0.20	15.9%	3	0.02	0.0%	0.10	7.0%
		Spoken	12	0.20	15.9%	1	0.01	0.0%	0.09	7.0%
ASR_C 13.7%	LIMSI1	Written	18	0.33	29.5%	19	0.24	17.9%	0.28	23.0%
		Spoken	18	0.33	29.5%	19	0.25	19.6%	**0.28**	**24.0%**
	UPC1	Written	12	0.22	20.5%	4	0.05	3.6%	0.13	11.0%
		Spoken	8	0.13	11.4%	2	0.03	1.8%	0.07	6.0%
	UPC2	Written	11	0.20	18.2%	4	0.03	1.8%	0.11	9.0%
		Spoken	10	0.21	20.5%	3	0.02	0.0%	0.10	9.0%

Table 10. Results for task T3, French Broadcast News, manual transcripts (68 factual questions and 32 definitional ones)

ASR	System	Questions	Factual			Definitional			All	
			#Correct	MRR	Acc	#Correct	MRR	Acc	MRR	Acc
ASR_A 11.0%	LIMSI1	Written	33	0.33	25.0%	19	0.47	37.5%	0.37	29.0%
		Spoken	32	0.33	25.0%	18	0.45	37.5%	0.37	29.0%
ASR_B 23.9%	LIMSI1	Written	25	0.29	25.0%	15	0.38	31.3%	0.32	27.0%
		Spoken	25	0.27	22.1%	13	0.35	31.3%	0.30	25.0%
ASR_C 35.4%	LIMSI1	Written	25	0.26	20.6%	13	0.33	28.1%	0.28	23.0%
		Spoken	24	0.25	19.1%	11	0.31	28.1%	0.27	22.0%

Accuracy goes down with increased word error rate giving a roughly 5% loss for ASR_B and ASR_C compared to ASR_A. It is interesting to note that the differences between ASR_B (WER 14%) and ASR_C (WER 24%) are negligible. The INAOE multi-ASR approach paid off by giving an overall result better than what was obtained by the same system on the best ASR only.

We notice that the impact of written vs spoken questions is similar than for manual transcriptions, with two systems taking an heavy loss and the others not showing a significant difference.

Four systems (2 from LIMSI and 2 from UPC) participated in the T2 (Spanish) task on manual transcripts and 3 systems (1 from LIMSI and 2 from UPC) on automatic transcripts.

On manual transcripts, the accuracy ranged from 36% (written questions and spoken questions) to 14% (written questions) and 17% (spoken questions). The differences between written questions and spoken questions is very low (from 0% to 3%). The same kind of behaviour is observed on the automatic transcripts tasks, with a loss due to the speech recognition errors and no significant difference between written and spoken questions.

Only 2 systems (both from LIMSI) participated in the T3 (French) task on manual transcripts and one (from LIMSI) on automatic transcripts.

On manual transcripts, the accuracy ranged from from 28% (both written and spoken questions) to 27% (written questions). There is no significant differences between spoken and written questions (0% to 1% loss). The results for automatic transcriptions show very little loss compared to the manual transcriptions except for the worst ASR.

The overall absolute results were worse this year compared to last year which points to a globally harder task. The question development method produces requests which qualitatively seem to be more different to what is found in the documents compared to questions built after reading the documents. In our opinion that method, while giving an harder problem, puts us closer to a real, usable application.

6 Conclusions

In this paper, the QAST 2009 evaluation has been described. Four groups participated in this track with a total of 86 submitted runs across 3 main tasks that included dealing with different languages (English, Spanish and French), different word error rates for automatic transcriptions (from 10.5% to 35.4%) and different question types (written and spoken questions). A novel question creation method has been tried succesfully to generate spontaneous spoken questions. Qualitatively, the questions were harder and more different to the formulations found in the documents compared to those produced by the traditional method of consulting the documents first. The method used this year gives an harder problem but we think that it is a more realistic one, putting us closer to a real, usable application.

Acknowledgments

This work has been jointly funded by the Spanish Ministry of Science (KNOW2 project - TIN2009-14715-C04-01 and TEXT-ENTERPRISE 2.0 project - TIN 2009-13391-C04-03) and OSEO under the Quaero program. We thank to Lori Lamel, Erik Bilinski, Manuel González and Pere Vilarrubia their help to the organisation and data generation.

References

1. Fiscus, J.: A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (rover). In: Proceedings 1997 IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara, CA, pp. 347–352 (1997)
2. Galliano, S., Geoffrois, E., Gravier, G., Bonastre, J.F., Mostefa, D., Choukri, K.: Corpus description of he ESTER Evaluation Campaign for the Rich Transcriptionof French Broadcast News. In: Proceedings of LREC 2006, Genoa, pp. 315–320 (2006)
3. Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.): CLEF 2006. LNCS, vol. 4730. Springer, Heidelberg (2007)
4. TC-Star (2004-2008), http://www.tc-star.org
5. Turmo, J., Comas, P.R., Ayache, C., Mostefa, D., Rosset, S., Lamel, L.: Overview of qast 2007. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 249–256. Springer, Heidelberg (2008)
6. Turmo, J., Comas, P.R., Rosset, S., Lamel, L., Moreau, N., Mostefa, D.: Overview of qast 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 314–324. Springer, Heidelberg (2009)
7. Voorhees, E.M., Buckland, L.L. (eds.): The Fifteenth Text Retrieval Conference Proceedings, TREC 2006 (2006)

GikiCLEF: Expectations and Lessons Learned

Diana Santos and Luís Miguel Cabral

Linguateca, Oslo node, SINTEF ICT, Norway
{Diana.Santos,Luis.M.Cabral}@sintef.no

Abstract. This overview paper is devoted to a critical assessment of GikiCLEF 2009, an evaluation contest specifically designed to expose and investigate cultural and linguistic issues in Wikipedia search, with eight participant systems and 17 runs. After providing a maximally short but self contained overview of the GikiCLEF task and participation, we present the open source SIGA system, and discuss, for each of the main guiding ideas, the resulting successes or shortcomings, concluding with further work and still unanswered questions.

1 Motivation

One of the reasons to propose and organize GikiCLEF (and the previous GikiP pilot [1]) was our concern that CLEF did not in general propose realistic enough tasks, especially in matters dealing with crosslingual and multilingual issues, both in topic/question creation and in the setups provided. In other words, while sophisticated from many points of view, CLEF setup was deficient in the attention paid to language differences (see e.g. [2,3]) or to the task definition [4,1].

While we all know in IR evaluation that laboratory testing has to be different from real life, and that a few topics or choices are not possible to validate a priori, but have to be studied after enough runs have been submitted and with respect to the pools and systems that were gathered[1], we wanted nevertheless to go some steps further, attempting to satisfy the following desiderata. GikiCLEF thus should:

1. provide a marriage of information needs and information source with real-life anchoring: and it is true that the man in the street does go to Wikipedia in many languages to satisfy his information needs;
2. tackle questions difficult both for a human being and for a machine: basically, we wanted a task with real usefulness, and not a task which would challenge systems to do what people don't want them to do. On the other hand, we wanted of course tasks that were possible to assess by (and satisfy) people, and not tasks that only computers could evaluate;
3. implement a context where different languages should contribute different answers, so that it would pay to look in many languages in parallel;
4. present a task that fostered the deployment of multilingual (and monolingual) systems that made use of comparable corpora.

[1] In fact, although this has been done for TREC – see [5,6] – it still remains to be done for CLIR or MLIA, although GridCLEF [7] is a significant step in this direction.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 212–222, 2010.

We also note that GikiCLEF was organized after a successful GikiP pilot which had already been meant as first step in these directions: GikiP, run in 2008, offered fifteen list questions to be solved in three language Wikipedias (Portuguese, German, and English), but had only three participants. As expounded in [8], we hoped that a larger contest could be organized that would foster research in useful tasks that required cultural awareness and were not based or centered around English alone.

Given that GikiCLEF 2009's setup and results have already been described in detail in the pre-workshop working notes [9], as well as being documented in its website,[2] we devote the current text to two main subjects: a presentation of SIGA as a reusable tool for new and related campaigns; and a discussion of whether GikiCLEF really managed to address and evaluate the task of "asking culturally challenging list questions to a set of ten different Wikipedias", presenting the achievements and shortcomings of what was in our opinion accomplished. We start in any case by offering a short description of the GikiCLEF task in order that this article be self-contained.

2 Very Brief Description of the Task

Systems participating in GikiCLEF were supposed to find, in several languages,[3] answers to questions that required or expected reasoning of some sort (often geographical, but also temporal and other).

In order to be considered as a correct answer, systems had to present it and a set of (Wikipedia) pages that justified it, in the eyes of a human being. Systems were thus invited to provide justification chains, in all the cases where the process of getting an answer involved visiting and understanding more than one Wikipedia page (see GikiCLEF's website for the exact submission format).

From the point of view of the assessment, this meant that, in order for the GikiCLEF setup to mirror a useful task, human assessors had to decide whether a given answer was (i) correct (by reading the pages or because they knew it) and (ii) justified (and, in that case, prior knowledge would not suffice).

Additionally, even if they knew better, assessors were required to "believe" Wikipedia, in the sense that even a wrong answer should be accepted as correct – according to the source, of course.

The extremely simple evaluation measures should only obey two constraints: One, the more languages the participant systems were able to provide answers in, the better. Two, systems should not be penalized if there were no answers in a particular language (Wikipedia). GikiCLEF scores were thus computed as the sum, for each language, of

[2] http://www.linguateca.pt/GikiCLEF/

[3] The GikiCLEF 2009 languages were: Bulgarian, Dutch, English, German, Italian, Norwegian – both Bokmål and Nynorsk –, Portuguese, Romanian and Spanish. A remark is in order concerning Norwegian: since it has two written standards, and Norwegians keep Wikipedia in two "parallel" versions, GikiCLEF covers nine languages but ten collections. Since both written standards of Norwegian were dealt equally in GikiCLEF, we will talk loosely of ten languages in what follows.

precision times the number of correct answers. For each language, the score was C*C/N (so that one had a score for de, pt, etc, as $C_{de} * C_{de}/N_{de}$, $C_{pt} * C_{pt}/N_{pt}$, etc.).[4]

In order to avoid machine translation problems – or even the lack of MT systems for any of the language (pairs) – the 50 questions were provided in all languages, for them to be on an equal footing. This was possibly the only unrealistic bit of the GikiCLEF setup, but let us stress that even for human beings the translation was not an easy task (again, see the website and the working notes paper for details). If we had relied on the participating systems having to invoke on their own MT for the topics (which had to be provided in different languages), we believe this would introduce a lot of uninteresting noise in the system.

Due to this choice, anyway, GikiCLEF can also be conceived as ten different evaluation contests (each asking questions in ONE language to a set of ten collections). So, the GikiCLEF evaluation has also provided results per language.

3 The SIGA System

SIGA[5] follows a similar structure as other systems such as DIRECT [10] or the one used in INEX [11], encompassing multiple user roles for different tasks. Different choices and privileges are thus in action for e.g. topic creation, run submission and validation, document pool generation, (cooperative) assessment, and computation and display of results. As new capabilities of SIGA we should mention the support for assessment overlap and subsequent conflict resolution process, both within the same language/collection, and across languages/collections.

To give a flavour of SIGA, we picked the assessment and the result computation facets. SIGA's assessment interface has three methods of navigation : (i) move to next/previous; (ii) move to next/previous in my list of assessments; (iii) move to next/previous item waiting to be assessed in my list of assessments.

As many important tasks were dependent on JavaScript (AJAX), the interface was made compatible with the most common browsers (IE and Mozilla). An example: when assessing an answer, and to minimize waiting time for the assessors, AJAX requests were used to preview documents answers and justifications, while assessing correctness and/or the justified property (which are two different actions in the interface).

Another feature of SIGA is that it allows inspection of the (individual and aggregated) results in several tables and graphics, based on the evaluation measures adopted by GikiCLEF, as can be seen in Figure 1. (We plan to allow for the customization of these measures in future versions.)

SIGA was released with the GNU GPL open source license and we aimed at easy installation. However, given that the system was primarily built to support GikiCLEF requirements, considerable work remains to be done in the following domains: support

[4] C stands for number of correct and justified answers provided by the system in that language, N for the total number of answers that the system came up with.

[5] SIGA stands for *SIstema de Gestão e Avaliação do GIKICLEF*, Portuguese for "Management and Evaluation System of GikiCLEF". The word *siga* means "Go on!" (imperative of verb *seguir*, "continue").

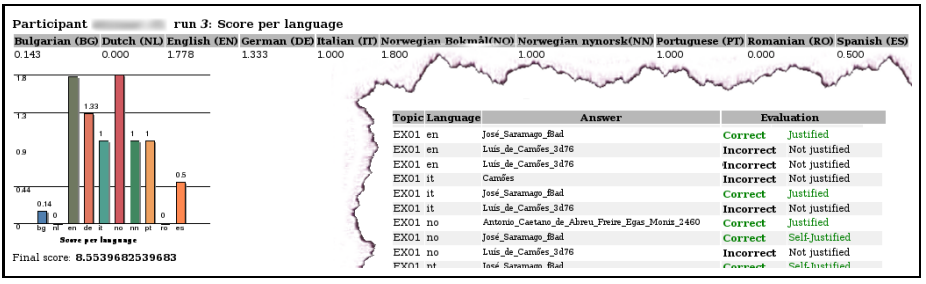


Fig. 1. SIGA in result mode: on the left, a graphic with language score; on the right, the assessment of each answer

for internationalization, easy addition of more metrics and plot solutions, and dealing with collections other than Wikipedia.

We have recently added a new functionality to SIGA, namely the possibility to try out new runs and provide corresponding additions to the pool for post-campaign experiments. This is extremely relevant for participants to do fine-grained error analysis and also to allow cooperative improvement of GikiCLEF resources.

In any case, it should be stressed that it is hard to do a system that remains useful for a long time when it deals with a dynamic resource such as Wikipedia. It is well known that Wikipedia has a steady growth, which may be accompanied by changes in format or structure. For example, differences in that respect were even noticeable among GikiCLEF languages. So, while SIGA currently allows to inspect answers (that is, Wikipedia pages, stripped of images and other links) in HTML, in XML[6] as well as in the current online version (last tested November 2009), changes in Wikipedia format and directives may occur so that future adaptation of SIGA may be required, as was incidentally the case when adapting WikiXML to GikiCLEF purposes.

Table 1. Sizes of different Wikipedia collections: For GikiCLEF, in addition to count the number of pages whose name starts by *Category* we provide also the number provided by the WikiXML conversion

	INEX Collection		GikiP collection		GikiCLEF collection		
Language	No. docs	No. cats	No. docs	No. cats	No. docs	No. cats	No. cats in Wikimedia
en	659,388	113,483	2,975,316	189,281	5,255,077	365,210	390,113
de	305,099	27,981	820,672	34,557	1,324,321	53,840	53,610
pt	–	–	286,296	22,493	830,759	51,001	48,761
nl	125,004	13,847	344,418	22,110	644,178	38,703	37,544
es	79,236	12,462	–	–	641,852	65,139	60,556

To provide some quantitative data on collection size, we show in Table 1 a comparison with two previous Wikipedia-based collections for evaluation, namely the ones

[6] Converted with the WikiXML tool created by the University of Amsterdam, available from http://ilps.science.uva.nl/WikiXML/

used in GikiP (from November 2006) and in the INEX collection [12], from January/February 2006. From INEX to GikiCLEF, it can be seen that Wikipedia grew up to ca. 800% for Spanish and English, as far as the number of documents is concerned. The number of categories has also grown considerably, up to almost 500% for Spanish.

4 Addressing the Crosslingual and Crosscultural Issue

Amassing Information Needs. In GikiP, most answers had been found in the three languages, therefore reducing the value of having a multilingual collection. So we decided to take the bull by the horns and heavily turn to culturally-laden questions, that is, questions about which one would expect a particular language or culture to display far more information than others.

In order to do that, we gathered a large organization committee with people from eight different countries/languages: there were Bulgarian, Dutch, German, Italian, Norwegian, Portuguese, Romanian and Spanish native speakers in the topic group, and we expressly requested that they came up with GikiCLEF topics that were not too global.

However, we had not foreseen that, by requiring people to choose topics of interest for their own language and culture, they would often choose those that their compatriots had carefully stored in the English Wikipedia as well, so that in fact the topic set became a sort of star with English as pivot. Table 2, borrowed and slightly modified from [13], displays the (current known) extent of the topics in the several GikiCLEF languages, by language/culture bias. One can see that most topics, no matter their cultural origin, had most hits in the English Wikipedia.

Table 2. Best languages per topic bias: in gray are the languages with the largest number of hits per topic. The rows describe the cultural topic bias as analysed by Nuno Cardoso, *none* meaning that no particular GikiCLEF language should a priori be best in it.

	total	bg	de	en	es	it	nl	nn, no	pt	ro
none	3	3	3	3	3	3	3	3	2	3
europe	6	1	4	5	1	2	2	1	2	1
bg	3	**3**	0	3	0	0	0	0	0	0
de	14	0	**10**	6	0	1	0	0	0	0
en	2	0	0	**1**	0	1	0	0	0	0
es	4	0	2	2	**1**	1	1	0	1	0
it	11	2	6	6	4	**7**	3	3	4	2
nl	6	1	1	2	0	0	**2**	0	0	0
nn, no	3	0	0	1	1	0	0	**1**	0	0
pt	2	0	2	1	0	0	0	0	**1**	0
ro	5	0	0	5	1	0	0	0	0	**2**

Guaranteeing Difficult, Non-trivial Questions. On the issue of finding user needs that required complex navigation and browsing in Wikipedia, therefore in need of automated help – that as far as we know is still not available for querying Wikipedia –, there was no doubt we succeeded.

The trouble might have been that the questions or topics were too difficult for systems as well, and thus GikiCLEF has been described as well ahead in the future. For more information on the topics and how they matched the collections, see again [13, 9]. There were nine topics for which no correct answer was returned.

The Added Value of Crosslinguality and Multilinguality Another issue was whether multilingual systems could get some value by using or reusing a comparable and parallel resource such as Wikipedia.

In one aspect, it is undeniably true that a bunch of participant systems were able, with this setup, to provide answers in languages they did not cover in any detail. This is advantageous because it shows that with a minimum work one can significantly widen the range of users one can satisfy, so we believe this should count as a GikiCLEF success.

Let us note this is not only a matter of following blindly language links from different language versions of Wikipedia (as it was almost always the case in GikiP): in fact, we were careful to provide a mechanism of crosslingual justification, in the sense that an answer was considered correct if it had a sibling which was justified. This was one could answer questions in Portuguese whose justification was only in Romanian or Bulgarian. This is obviously an added value of using other languages, even only in a monolingual setup.[7]

However, the value of processing different languages instead of English was not at all ascertained, as already described in subsection 4 and as we will show further in the next section.

Table 3. Participants in GikiCLEF 2009: *Langs.* stands for languages of participation, *NL* stands for native language of the system, if not all equally treated.

Name	Institution	System name	Langs.	NL
Ray Larson	University of California, Berkeley	cheshire	all	en
Sven Hartrumpf &	FernUniversität in Hagen &	GIRSA-WP	all	de
& Johannes Leveling	& Dublin City University			
Iustin Dornescu	University of Wolverhampton	EQUAL	all	en
TALP Research Center	Universitat Politécnica de Catalunya	GikiTALP	en,es	en,es
Gosse Bouma & Sergio Duarte	Information Science,	JoostER	du,es	du,es
&Sergio Duarte	University of Groningen			
Nuno Cardoso et al.	GREASE/XLDB, Univ. Lisbon	GreP	all	pt
Adrian Iftene et al.	Alexandru Ioan Cuza University	UAICGIKI09	all	all
Richard Flemmings et al.	Birkbeck College (UK) &	bbk-ufrgs	pt	pt
	UFRGS (Brazil)			

Actual Participation and Subsequent Answer Pool. In fact, GikiCLEF 2009 was not able to provide a setup where seriously processing languages other than English provided a considerable advantage. The particular group of participants in GikiCLEF

[7] A pedantic user could wish to know in each language was it actually justified, but most users asking for list questions would be satisfied knowing that the system had justified the answer some way.

(see Table 3) should also in a way be held responsible for this conclusion, as we proceed to explain.

In fact, an unexpected detail in GikiCLEF that also conspired against our initial goals was that there were very few participating groups from non-English languages, which meant that the pool (the results we actually got) are much better in English. This is hardly surprising if the bulk of the processing was made in English. Figure 4 shows this clearly.

Let us stress this here: Our pool does not necessarily mean that the answers to the questions were better answered by the English Wikipedia, no matter its larger size. It is also equally a consequence of the particular group of participant systems.

More concretely, we emphasize that there were no pure Bulgarian, Italian, Norwegian or Romanian participants, which means that most answers got in those languages came from following links from other languages.[8] Likewise, there was only one Dutch and one German participant, while Spanish and Portuguese, although having more devoted participants, were not able to significantly gather more answers because of that, given that some of these dedicated systems had hardly any correct answer to contribute to the pool.

This means that, in fact and although expected otherwise, what GikiCLEF 2009 amounted to was to ascertain how well systems can answer multilingual/multicultural questions by simply processing English and following the Wikipedia links (although some systems tried the reverse as well). This is a relevant and interesting issue in itself, but it must be emphasized that it is very far from the research question we had in the first place.

5 Was GikiCLEF in Vain?

The final balance we do is therefore mixed. Although the initial purpose was not achieved, several resources were gathered and deserve further study. We have also laid the foundations for organizing future venues which are similar in spirit, as well as offered a system that allows easy gathering of further empirical data.

The first and obvious lesson learned was that generalization or extension from a pilot is not free from danger. While we may have correctly diagnosed that GikiP was not interesting enough because one could get the very same data by processing only one language, the suggested fix had the opposite effect, by effectively electing English as the best language to win at GikiCLEF.

But note: we came to realise as well that GikiCLEF was very far from a realistic situation. Quite the opposite, it will strike anyone who gives it some thought that the topic collection is very far from representing any single individual or the usual needs or interests of one particular community: it is a mix of a set of ten or more individuals – each of

[8] This is a truth with modifications, since the UAICGIKI09 system actually processed all languages in parallel. However, its contribution to the pool was rather poor. Note also that we are not interested in the country of origin of the researchers but simply whether their systems treated in a special and knowledgeable way a particular language. When systems participated in a partially interactive run, it is even more difficult to decide what languages really were independently natively processed.

Table 4. Results in GikiCLEF 2009: The last row indicates how many participants per language, and the last column the number of languages tried in that run. Eight runs opted for all (10) languages, four tried solely 2 languages, and five one only.

System	bg	de	en	es	it	nl	nn	no	pt	ro	Score	L
EQUAL	9.757	25.357	34.500	16.695	17.391	21.657	9.308	17.254	15.515	14.500	181.933	10
GreP	6.722	12.007	13.657	11.115	8.533	8.258	9.557	11.560	7.877	6.720	96.007	10
Cheshire	1.091	9.000	22.561	4.923	11.200	9.132	3.368	7.043	4.891	7.714	80.925	10
GIRSA 1	1.333	3.125	1.800	3.000	2.250	2.250	2.000	3.000	3.000	3.000	24.758	10
GIRSA 3	3.030	3.661	1.390	2.000	1.988	1.798	3.064	2.526	2.250	1.684	23.392	10
GIRSA 2	2.065	1.540	0.938	1.306	1.429	1.299	1.841	1.723	1.350	1.029	14.519	10
JoostER 1	—	—	1.441	—	—	0.964	—	—	—	—	2.405	2
GTALP 3	—	—	1.635	0.267	—	—	—	—	—	—	1.902	2
GTALP 2	—	—	1.356	—	—	—	—	—	—	—	1.356	1
GTALP 1	—	—	0.668	0.028	—	—	—	—	—	—	0.696	2
bbkufrgs 1	—-	—	—	—	—	—	—	—	0.088	—	0.088	1
UAICG 2	0.000	0.002	0.002	0.006	0.002	0.002	0.000	0.002	0.002	0.000	0.016	10
bbkufrgs 2	—	—	—	—	—	—	—	—	0.012	—	0.012	1
UAICG 1	—	—	—	0.006	—	—	—	—	—	0.000	0.006	2
UAICG 3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	10
bbkuf 3	—	—	—	—	—	—	—	—	0.000	—	0.000	1
JoostER 2	—	—	—	0.000	—	—	—	—	—	—	0.000	1
Runs	8	8	12	12	8	9	8	8	11	9		

whom has probably tried to come up with diverse questions, and not even his or her own real interests.

So, in hindsight, we may say that the GikiCLEF topic set was fairly unrealistic and that we had better concentrate on a specific area or kind of user to really test systems for a particular application. Of course, this is often something that is hard to do in the context of a general evaluation: if one wants to evaluate tasks and have a broad participation, one cannot concentrate in a too narrow (but realistic) set of users: those interested in Romanian literature, for example.

Rephrasing the problem: we had too few questions of each kind of subject / language, together with the fact that a miriad of other factors also played a non-despicable role. If we had 50 topics each of interest for one given language/culture/community, then we might be able to smooth the role of individual differences. But – just to give a striking example – there was only one question that was specifically related to a Portuguese-speaking culture (about Brazilian coastal states). So, a system working only or primarily in Portuguese would have (probably) advantage for that topic only, while for 25 topics it would have had absolutely no answer in Portuguese (according to [14]). In other words, such a system in GikiCLEF had the possibilities of getting a good score halved from the start. And the same unprivileged situation applied to Bulgarian or Dutch, even if they had 3 or 4 biased topics in the total 50.

As already noted in e.g. [15], as far as we know there has never been an investigation on the role/weight of language/culture in previous CLEF contests. Adhoc tracks were often designed to have answers in most languages, but it was hardly discussed whether

these answers[9] were similar or distinct, in the sense of representing the "same" information (but just discussed or presented in a different way). So, we have no idea whether multilingual search was beneficial in those tracks, neither how contrived and/or general the topics had to be in order to be chosen, and it may well be that the conclusions and failures reported for GikiCLEF apply to these other setups as well.

As reported in [9], organizing GikiCLEF in 2009 allowed us to amass a significant number of resources as far as judgements are concerned, of which the most important were possibly the 1,009 correct and justified answers for 50 topics in 10 Wikipedias (1,621 if we count only correct, not necessarily justified). But we know that a lot of work still remains to be done to have a good evaluation resource.

All resources have been joined in the GIRA package, available from http://www.linguateca.pt/GikiCLEF/GIRA/, which we expect to improve in the future by delivering further versions.

In fact, we think it is important and worth while to enhance this data with actual work done by users genuinely interested in the particular topics and with native or good competence in the several languages, in order to get a better overview of the knowledge that is included in Wikipedia(s) and the upper limit that systems could get at.

Another course of action relatively easy to implement would be to provide recall measures based on the improved pool, as suggested for example by Iustin Dornescu [16].

Also, one should be able to gather a better overview of the relative difficulty of the different questions, if we were able to get this job done by human volunteers, as Ray R. Larson [17] one of the participants started to do. For example, the pool is uneven even due to the fact that the cheshire system, for lack of time, only delivered answers to the first 22 topics.

Note that there are two difficulties with the two suggestions just made, though: (i) GikiCLEF topics were most often than not meant to be discovery topics, that is, the topic owner did not know all answers beforehand, so we may never be sure about absolute recall; and (ii) many questions may require huge human labour to be answered.

Incidently, although the main target of GikiCLEF was open list questions, some closed questions were inadvertently included, and also even some with only one answer. This second issue, however, in our opinio only makes GikiCLEF more realistic, in the sense that an ordinary questioner might not know that there was a unique answer.[10]

A final issue which in our opinion deserves further study, is to consider more carefully how far the "same" answer can be said to be given/present in different languages. In addition to the already mentioned fact mismatches reported e.g. in [1,8], other more subtle problems concern categories: we enforced category type checking – which was

[9] Which were documents and not strings, that is, not precise answers.

[10] Only to reject would be those questions where that was presupposed in the question formulation, such as "Who is the unique...", which we declared as uninteresting fro GikiCLEF. But we are aware that this was just an evaluation contest limitation, obviously similar (and not list) questions involving some kind of ranking are often equally interesting and important to answer, such as who was the first, which is highest, and so on, and should not be harder or different to anwer by GikiCLEF systems, were it nor for the fact that often these properties are also mentioned in the text on an entry, and have this easy shortcuts.

often a problem for assessors as reported in [9] – but in some cases categories were not alignable across languages. For example, in some languages the category "ski resorts" was not available, even if all information was duly described in the corresponding village or mountain pages.[11] Also, cases where lexicalization was different – and thus lexical gaps exist – provide obvious problems for language linking. So, a study of the misalignability of the different Wikipedias is relevant in itself, not only for GikiCLEF-like systems, but also for the large number of other NLP systems out there who rely on Wikipedia as a multilingual or translation resource.

In a nutshell, we have made the obvious discovery that, if one wants to go beyond a quite basic simplicity level, one has to deal with all philosophical and intriguing questions that natural language understanding poses.

Acknowledgements

We are grateful to Nuno Cardoso for preparing the colections, and to the remaining organizers – Sören Auer, Gosse Bouma, Iustin Dornescu,Corina Forascu, Pamela Forner, Fredric Gey, Danilo Giampiccolo, Sven Hartrumpf, Katrin Lamm, Ray Larson, Johannes Leveling, Thomas Mandl, Constantin Orasan, Petya Osenova, Anselmo Peñas, Alvaro Rodrigo, Julia Schulz, Yvonne Skalban, and Erik Tjong Kim Sang – for hard work, supportive feedback and unfailing enthusiasm.

We also thank the larger set of further assessors – including the organizers, but also Anabela Barreiro, Leda Casanova, Luís Costa, Ana Engh, Laska Laskova, Cristina Mota, Rosário Silva and Kiril Simov – who performed the assessment, as well as Paula Carvalho and Christian-Emil Ore, who helped in an initial phase suggesting Portuguese and Norwegian-inspired topics, respectively.

Our work was done under the scope of the Linguateca project, jointly funded by the Portuguese Government, the European Union (FEDER and FSE), under contract ref. POSC/339/1.3/C/NAC, UMIC and FCCN. We also gratefully acknowledge support of the TrebleCLEF Coordination Action, ICT-1-4-1 Digital libraries and technology-enhanced learning (Grant agreement: 215231).

References

1. Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J., Skalban, Y.: GikiP at GeoCLEF 2008: Joining GIR and QA forces for querying Wikipedia. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 894–905. Springer, Heidelberg (2009)
2. Santos, D., Rocha, P.: The key to the first CLEF in Portuguese: Topics, questions and answers in CHAVE. In: Peters, C., Clough, P., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B. (eds.) CLEF 2004. LNCS, vol. 3491, pp. 821–832. Springer, Heidelberg (2005)
3. Santos, D., Cardoso, N.: Portuguese at CLEF 2005: Reflections and Challenges. In: Peters, C. (ed.) CLEF 2005. LNCS, vol. 4022, pp. 1007–1010. Springer, Heidelberg (2006)

[11] In that case we had to relax the type checking constraint during assessment and conflict resolution modes.

4. Santos, D., Costa, L.: QolA: fostering collaboration within QA. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 569–578. Springer, Heidelberg (2007)
5. Zobel, J.: How Reliable Are the Results of Large-Scale Information Retrieval Experiments? In: SIGIR'98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 307–314. ACM, New York (1998)
6. Voorhees, E.M., Buckley, C.: The effect of topic set size on retrieval experiment error. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 316–323 (2002)
7. Ferro, N., Harman, D.: CLEF 2009: Grid@CLEF Pilot Track Overview. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 553–566. Springer, Heidelberg (2010)
8. Santos, D., Cardoso, N.: GikiP: Evaluating geographical answers from Wikipedia. In: Proceedings of the 5th Workshop on Geographic Information Retrieval (GIR 2008), Napa Valley, CA, USA, pp. 59–60 (2008)
9. Santos, D., Cabral, L.M.: GikiCLEF: Crosscultural issues in an international setting: asking non-English-centered questions to Wikipedia. In: Borri, F., Nardi, A., Peters, C. (eds.) Cross Language Evaluation Forum: Working notes for CLEF (2009)
10. Dussin, M., Ferro, N.: Direct: applying the dikw hierarchy to large-scale evaluation campaigns. In: Larsen, R.L., Paepcke, A., Borbinha, J.L., Naaman, M. (eds.) Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries, pp. 424–424. ACM, New York
11. Lalmas, M., Piwowarski, B.: INEX 2006 relevance assessment guide. In: INEX 2006 Workshop Pre-Proceedings, pp. 389–395 (2006)
12. Denoyer, L., Gallinari, P.: The Wikipedia XML corpus. ACM SIGIR Forum 40, 272–367 (2006)
13. Cardoso, N.: GikiCLEF topics and Wikipedia articles: did it blend? In: CLEF2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)
14. Cardoso, N., Batista, D., Lopez-Pellicer, F., Silva, M.J.: Where in the Wikipedia is that answer? The XLDB at the GikiCLEF 2009 task. In: Borri, F., Nardi, A., Peters, C. (eds.) Cross Language Evaluation Forum CLEF 2009 Workshop (2009)
15. Santos, D., Cardoso, N.: Portuguese at CLEF. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, pp. 1007–1010. Springer, Heidelberg (2006)
16. Dornescu, I.: EQUAL Encyclopaedic QA for Lists. In: CLEF2009 Workshop, Corfu, Greece (2009)
17. Larson, R.R.: Interactive probabilistic search for GikiCLEF. In: Borri, F., Nardi, A., Peters, C. (eds.) Cross Language Evaluation Forum: Working notes for CLEF 2009, Corfu, Greece (2009)

NLEL-MAAT at ResPubliQA

Santiago Correa, Davide Buscaldi, and Paolo Rosso

NLE Lab, ELiRF Research Group, DSIC,
Universidad Politécnica de Valencia, Spain
{scorrea,dbuscaldi,prosso}@dsic.upv.es
http://www.dsic.upv.es/grupos/nle

Abstract. This report presents the work carried out at *NLE Lab* for the *QA@CLEF-2009* competition. We used the *JIRS* passage retrieval system, which is based on redundancy, with the assumption that it is possible to find the response to a question in a large enough document collection. The retrieved passages are ranked depending on the number, length and position of the question n-gram structures found in the passages. The best results were obtained in monolingual English, while the worst results were obtained for French. We suppose the difference is due to the question style that varies considerably from one language to another.

1 Introduction

An open-domain *Question Answering* (*QA*) system can be viewed as a specific *Information Retrieval* (*IR*) system, in which the amount of information retrieved is the minimum amount of information required to satisfy a user information need expressed as a specific question, e.g.: "Where is the Europol Drugs Unit?". Many *QA* systems are based on *Passage Retrieval* (*PR*) [6,4]. A *PR* system is an *IR* system that returns parts of documents (passages) instead of complete documents. Their utility in the *QA* task is based on the fact that in many cases the information needed to answer a question is usually contained in a small portion of the text [3].

In the 2009 edition of *CLEF*, the competition *ResPubliQA*[1] has been organized, a narrow domain *QA* task, centered on the legal domain, given that the data is constituted by the body of *European Union* (*EU*) law. Our participation in this competition has been based on the *JIRS*[2] open source *PR* system, which has proved to be able to obtain better results than classical *IR* search engines in the previous open-domain *CLEF QA* tasks [1]. In this way we desired to evaluate the effectiveness of this *PR* system in this specific domain and to check our hypothesis that most answers usually are formulated similarly to questions, in the sense that they contain mostly the same sequences of words. In the next section, we describe the characteristics of the task; furthermore, Sect. 3 and 4

[1] For more information about the competition ResPubliQA@CLEF-2009, refer to page: http://celct.isti.cnr.it/ResPubliQA/

[2] http://sourceforge.net/projects/jirs/

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 223–228, 2010.

explain the main concepts of *JIRS* (*Java Information Retrieval System*) system and we discuss how it has been applied in solving the problem; in Sect. 5 we present the results and finally in Sect. 6 we draw some conclusions.

2 Multiple Language Question Answering Task

In this task, the system receives as input natural language questions about European law, and the system should return a paragraph containing the response from the document collection. This constitutes an important difference with respect to previous *QA* tasks where an exact answer had to be extracted or generated by the system. For this reason we employed just the *JIRS* system instead of the complete *QUASAR QA* system we developed for previous QA@CLEF participations [2].

The document collection is a subset of the *JRC-Acquis corpus* [3] , containing the complete *EU* legislation, including texts between the years 1950 to 2006 (in total 10,700 documents); these documents have been aligned in parallel and were made available to the participants in the following languages: Bulgarian, Dutch, English, French, German, Italian, Portuguese, Romanian and Spanish. The corpus is encoded in *XML* format according to the *TEI guidelines*[4] . Each document has a title and is subdivided into paragraphs, each one marked with the "<p>" tag. The test set is composed of 500 questions that must be analyzed by the systems to return a paragraph that contains the answer to the formulated question.

3 The Passage Retrieval Engine JIRS

Many passage retrieval systems are not targeted to the specific problem of finding answers, due to the fact that they only take into account the keywords of the question to find the relevant passages. The information retrieval system *JIRS* is based on n-grams (an n-gram is a sequence of n adjacent words extracted from a sentence or a question) instead of keywords. *JIRS* is based on the premise that in a large collection of documents, an n-gram associated with a question must be found in the collection at least once.

JIRS starts searching the candidate passage with a standard keyword search that retrieves an initial set of passages. These passages are ranked later depending on the number, position and length of the question n-grams that are found in the passages. For example: suppose you have a newspaper archive, using the JIRS system and based on these documents you will find the answer to the question: "Who is the president of Colombia?". The system could retrieve the following two passages: "... Álvaro Uribe is the president of Colombia ..." and "...Giorgio Napolitano is the president of Italy...". Of course, the first passage should have more relevance as it contains the 5-gram "is the president of

[3] http://wt.jrc.it/lt/Acquis/
[4] http://www.tei-c.org/Guidelines/

Colombia", while the second passage contains only the 4-gram "is the president of". To calculate the n-gram weight of each passage, first of all we need to identify the most relevant n-gram and assign to it a weight equal to the sum of the weights of its terms. The weight of each term is set to:

$$w_k = 1 - \frac{log(n_k)}{1 + log(N)} \tag{1}$$

Where n_k is the number of passages in which the term appears and N is the total number of passages in the system.

The similarity between a passage d and a question q is determined by:

$$Sim(d,q) = \frac{\sum_{j=1}^{n} \sum_{x \in Q} h(x, D_j)}{\sum_{j=1}^{n} \sum_{x \in Q} h(x, Q_j)} \tag{2}$$

Where $h(x, D_j)$ returns a weight for the j-gram x with respect to the set of j-grams (D_j) in the passage:

$$h(x, D_j) = \begin{cases} \sum_{k=1}^{j} w_k & \text{if } x \in D_j \\ 0 & otherwise \end{cases} \tag{3}$$

A more detailed description of the system *JIRS* can be found in [2].

4 Adaptation of JIRS to the Task

The data had to be preprocessed, due to the format of the collection employed in *ResPubliQA* competition, a subset of the *JRC-ACQUIS* Multilingual Parallel corpus, this corpus containing the total body of *European Union* (*EU*) documents, of mostly legal nature. In particular, the subset is constituted by documents of 9 out of 22 languages. It consists of approximately 10,700 parallel and aligned documents per language. The documents cover various subject domains: law, politics, economy, health, information technology, agriculture, food and more.

To be able to use the *JIRS* system in this task, the documents were analyzed and transformed for proper indexing. Since *JIRS* uses passages as basic indexing unit, it was necessary to extract passages from the documents. We consider any paragraph included between <p> tags as a passage. Therefore, each paragraph was labeled with the name of the containing document and its paragraph number.

Once the collection was indexed by *JIRS*, the system was ready to proceed with the search for the answers to the test questions. For each question, the system returned a list with the passages that most likely contained the answer to the question, according to the *JIRS* weighting scheme. The architecture of the monolingual *JIRS*-based system is illustrated in Fig. 1. In an additional experiment, we used the parallel collection to obtain a list of answers in different languages (Spanish, English, Italian and French). The idea of this approach is based on the implementation of 4 monolingual *JIRS*-based systems, one for each

language, which will have as input the set of questions in the respective language. For this purpose we used a tool (Google Translator[5]) to translate the entire set of questions into the same language. Later choosing as the best answer the one that obtained the best score by the system and subsequently taking the identifier of each paragraph (answer) for retrieving the aligned paragraph in the target language. The architecture of the multilingual *JIRS*-based system is illustrated in Fig. 2.

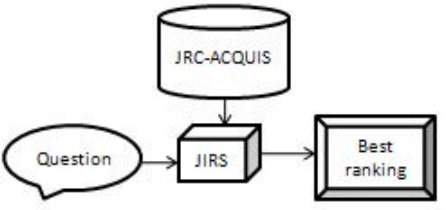


Fig. 1. Architecture of NLEL-MAAT monolingual system

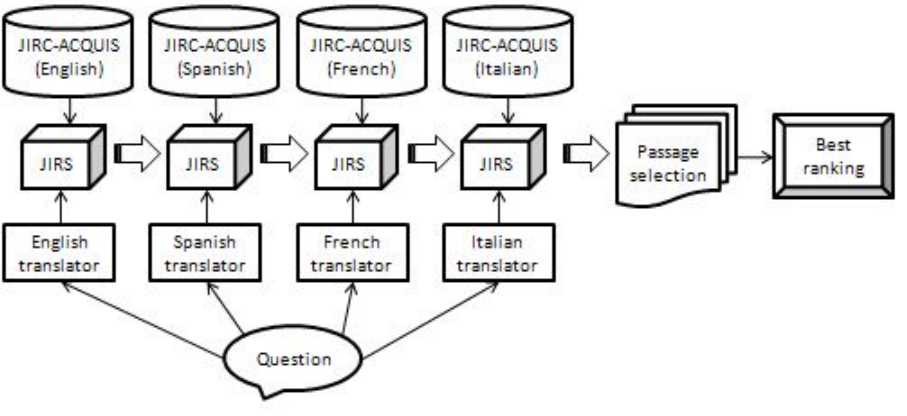


Fig. 2. Architecture of NLEL-MAAT multilingual system

5 Results

We submitted four "pure" monolingual runs for the following languages: English, French, Italian and Spanish, and in an additional experiment we exploited the parallel corpus to produce a monolingual Spanish run. This experiment consisted in searching the question in all languages, and selecting the passage with the highest similarity; finally, the returned passage was the Spanish alignment of this best passage. In Table 1 we show the official results for the submitted runs [5].

[5] http://www.google.com/language_tools?hl=en

Table 1. Results for submitted runs to ResPubliQA, Ans.: Answered, Unans.: Unanswered, A.R.: Answered Right, A.W.: Answered Wrong, U.R.: Unanswered Right, U.W.: Unanswered Wrong, U.E.: Unanswered Empty, Accuracy: Accuracy measure, c@1

Task	Ans.	Unans.	A.R.	A.W.	U.R.	U.W.	U.E.	Accuracy	c@1
en-en	498	2	287	211	0	0	2	0.57	0.58
fr-fr	489	11	173	316	0	0	11	0.35	0.35
es-es	495	5	173	322	0	0	5	0.35	0.35
it-it	493	7	256	237	0	5	2	0.51	0.52
es-es2	466	34	218	248	0	0	34	0.44	0.47

From Fig. 3 we can see that the result obtained in English were particularly good, while in French and Spanish the percentage of wrong answers is very high. We did not expect this behavior for Spanish, since *JIRS* was developed specifically for the Spanish *QA* task. Maybe the poor behavior was due to the peculiarity of the *JRC Acquis* corpus, containing, for instance, many anaphoras. On the other hand, we expected the French to be the language in which the system obtained the worst results because of the results of the system at previous QA competitions. The Spanish-multilingual approach allowed to reduce the wrong answers by 23% (from 322 to 248) and increase the number of right ones by 26% (from 173 to 218).

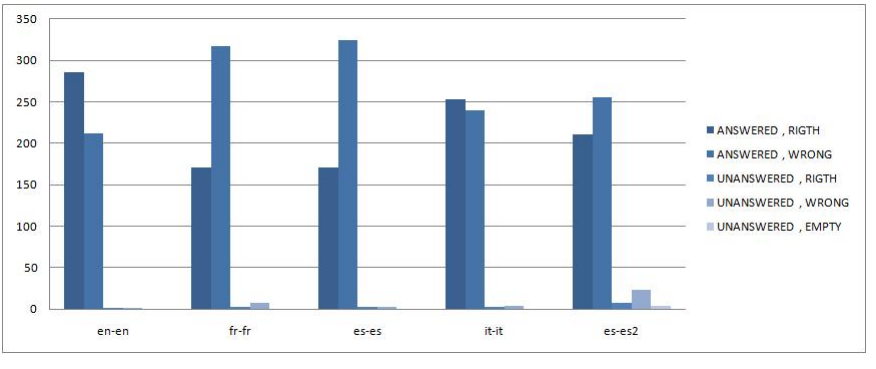


Fig. 3. Comparative graph for all the sumitted runs

6 Conclusions

The difference between the best results (for English) and the worst ones (for French) is 22 percent points in accuracy. This may reflect the different way of formulating questions in each language. In a comparison with other teams' results, we obtained excellent results, proof of this is the best rating in three of the four tasks we participated in. The only task in which the NLEL-MAAT

system was not ranked first is the monolingual task en-en. However, the system ranked second with just a difference of 0.03 in the $c@1$ measure and 0.04 in the *Accuracy* measure. Moreover it is also important to note that due to the language independence of *JIRS* we have participated and obtained very good results in all the tasks. Due to the improvement obtained using the parallel data set (es-es2 task) with respect to the Spanish monolingual task (es-es), we plan to employ this approach also for the other languages.

Acknowledgments. The work of the first author was made possible by a scholarship funded by Maat Gknowledge in the context of the project with the Universidad Politécnica de Valencia Módulo de servicios semánticos de la plataforma G. We also thank the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 research project.

References

1. Buscaldi, D., Gómez, J.M., Rosso, P., Sanchis, E.: N-Gram vs. Keyword-Based Passage Retrieval for Question Answering. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS (LNAI), vol. 4730, pp. 377–384. Springer, Heidelberg (2007)
2. Buscaldi, D., Rosso, P., Gómez, J.M., Sanchis, E.: Answering questions with an n-gram based passage retrieval engine. Journal of Intelligent Information Systems (82):(Online First, 2009) ISSN: 0925-9902 (Print) 1573-7675 (Online), doi: 10.1007/s10844-009-0082-y (2009)
3. Callan, J.P.: Passage-level Evidence in Document Retrieval. In: SIGIR 1994: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 302–310. Springer, New York (1994)
4. Neumann, G., Sacaleanu, B.: Experiments on Robust NL Question Interpretation and Multi-layered Document Annotation for a Cross-Language Question/Answering System. In: Peters, C., Clough, P., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B. (eds.) CLEF 2004. LNCS, vol. 3491, pp. 411–422. Springer, Heidelberg (2005), doi:10.1007/11519645_41
5. Pñas, A., Forner, P., Sutcliffe, R., Rodrigo, Á., Forascu, C., Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P.: Overview of respubliqa 2009: Question answering evaluation over european legislation. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece (2009)
6. Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G.: Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. In: SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Cnference on Research and Development in Information Retrieval, pp. 41–47. ACM, New York (2003)

Question Answering on English and Romanian Languages

Adrian Iftene[1], Diana Trandabăț[1,2], Alex Moruz[1,2],
Ionuț Pistol[1], Maria Husarciuc[1], and Dan Cristea[1,2]

[1] UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
[2] Institute of Computer Science, Romanian Academy Iasi Branch
{adiftene,dtrandabat,amoruz,ipistol,
mhusarciuc,dcristea}@info.uaic.ro

Abstract. 2009 marked UAIC1's fourth consecutive participation at the QA@CLEF competition, with continually improving results. This paper describes UAIC's QA systems participating in the Ro-Ro and En-En tasks. Both systems adhered to the classical QA architecture, with an emphasis on simplicity and real time answers: only shallow parsing was used for question processing, the indexes used by the retrieval module were at coarse-grained paragraph and document levels, and the answer extraction component used simple pattern-based rules and lexical similarity metrics for candidate answer ranking. The results obtained for this year's participation were greatly improved from those of our team's previous participations, with an accuracy of 54% on the EN-EN task and 47% on the RO-RO task.

1 Introduction

In 2009, the QA@CLEF track was called ResPubliQA[2]. The structure and the aims of the task remain almost the same as in previous years: given a pool of 500 independent questions in natural language, participating systems must return an answer for each question. The main difference from past editions comes from the fact that the document collection for 2009 was the JRC-Acquis corpus. Other changes influencing the development of this year's systems are the facts that the question types have changed, and that the answers were no longer expected be exact answer, but paragraphs extracted from the JRC Acquis corpus containing the correct answers.

Preparing the 2009 competition, we continued to improve our system built for the 2008 QA@CLEF edition [2], focusing on reducing the running time and increasing the performances. We indexed the new corpus at both paragraph and document level, and when looking for potential candidate answers, we kept both types of returned snippets: if the search for the answer in paragraph snippets was unsuccessful, we tried to identify the answer using the snippets returned by the document level index.

The best system in participating in this year's challenge for the Romanian language [4] showed a very good performance compared to the rest of the runs. This is a system

[1] University "Al. I. Cuza" of Iasi, Romania.
[2] ResPubliQA: http://celct.isti.cnr.it/ResPubliQA/

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 229–236, 2010.

that uses a sophisticated similarity based model for paragraph ranking, question analysis, classification and regeneration of the question, classification of paragraphs and consideration of the EUROVOC terms associated to each document. For English, the best runs produced paragraph rankings considering matching n-grams between question and paragraphs [1]. This retrieval approach seems to be promising, since combined with paragraph validation filters it achieved the best score [6] in English.

The general system architecture is described in Section 2, while Section 3 is concerned with presentation of the results. The last Section discusses the conclusions and further work envisaged after our participation in QA@CLEF 2009.

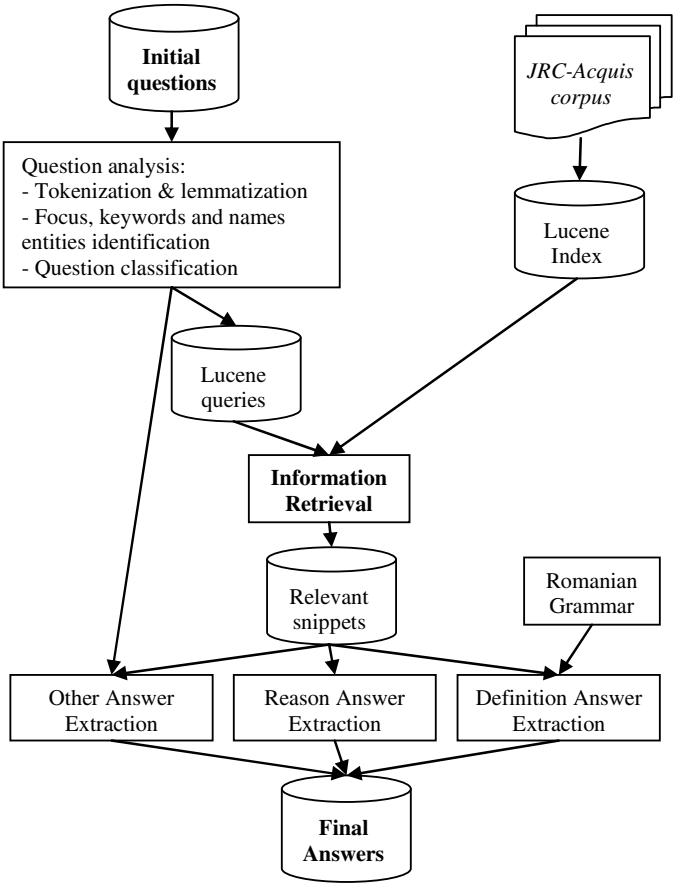


Fig. 1. UAIC system used in ResPubliQA

2 Architecture of the QA System

The architecture of the system we used in the ResPubliQA [5] track is presented in Figure 1. Similarly to last year's system, we eliminated many pre-processing modules in order to obtain a real-time system. Another feature that we used from last year's

system is a Romanian grammar for identifying different definition types [3], but with major improvements and adaptation for the juridical domain. The main differences from last year's participation are detailed in the following sections.

2.1 Corpus Pre-processing

The JRC-Acquis corpus is a collection of juridical documents in XML format, with each paragraph numbered. Because of the official nature of the documents, the corpus was in a very well organized structure, making unnecessary any additional cleaning.

2.2 Question Analysis

This step is mainly concerned with the identification of the semantic type of the answer (expected answer type). A specialized module identifies the question focus, the question type and a set of relevant keywords. The question analyzer performs the following steps:

i. NP-chunking and Named Entity extraction;
ii. Question focus identification (the most important word in the question, the clue for determining the answer type);
iii. Answer type identification;
iv. Question type inferring;
v. Identification of the keywords in the sentence that, together with the NPs and named entities, are to be used by the query generator.

The question analysis was performed using the module we already developed for the previous competitions, the major change being the question type identification. This year, seen the semantic nature of the new corpus and the possible interrogations over it, a new partitioning of question types was considered: factoid, definition, purpose, reason and procedure. Thus, the development questions provided by the organizers were used to learn how to identify among the new question types.

2.3 Index Creation and Information Retrieval

The purpose of this module is to retrieve the relevant snippets of text for every question. For this task we used the Lucene[3] indexing and search tools. A brief description of the module is given below:

i) Query creation
Queries are formed based on the question analysis. They are made up of the sequences of keywords we previously identified, which are modified using some Lucene operators, such as score boosting (the "^" operator, followed by a positive integer), fuzzy matching (the "~" operator, followed by a value between 0 and 1) and the "exclusive or" operator (symbolized by words between parentheses). As a rule of thumb, triggered by empirical observations, the score for the question keyword is boosted by a factor of 2 (^2), the score for the named entities in the query is boosted by a factor of 3 (^3), and, in the case of words that are not in a lemma form, we use

[3] Lucene: `http://lucene.apache.org/`

the "exclusive or" operator between the surface and the lemma forms (the inflected form being emphasized by boosting it by a factor of 2).

As Romanian is a heavily inflected language, in order to avoid using all of the inflected forms of a given word, and also to avoid lemmatizing the entire corpus, we have used the fuzzy match operator, which searches for words that are similar, up to a given degree, to the query word. Based on test performed over the training set, we have found that the value which gives the best results is a similarity score of 0.7. For example, in the case of the question "*La ce se referă rezoluția Consiliului despre educația copiilor lucrătorilor care se mută?"* (En: "*What is the scope of the Council resolution with regard to the children of moving workers?*"), the query is:

```
RO:(referă^2 referi) rezoluția~0.7 Consiliului^3
educația~0.7 copiilor~0.7 lucrătorilor~0.7 (mută^2 muta)

EN:(refer^2 refers) resolution~0.7 Council^3 education~0.7
children~0.7 workers~0.7 (move^2 moving moves)
```

ii) Index creation

The index was created using the XML files in the JRC-Aquis corpus. We have created two indexes, one at paragraph level and one at document level. The paragraph index is more precise in terms of relevant text, and is preferred for snippet extraction. If, however, the answer is not found in the paragraph index, the query is applied to the document index instead.

iii) Relevant snippet extraction

Using the queries and the indexes we used the Lucene search engine to extract a ranked list of snippets for every question as possible answer candidates.

2.4 Answer Extraction

In the 2009 year's track, we used the FACTOID question answer extraction module of the last year system, which was refined by including sub-types (person, organization, count, measure, temporal, etc.). We have built special modules to extract answers for questions of type DEFINITION and REASON. Simple pattern matching methods using rules learned from the training data were used for the other question types (PURPOSE and PROCEDURE).

Our algorithm for answer extraction is based on several heuristics that try to find the best answer from available candidates. Examples of the heuristics used are:

- the paragraph contains the question focus;
- the paragraph contains (at least) some of the named entities (directly proportional to the number of these name entities);
- if the question answer type is Person, Organization, Date, Measure, we try to identify these types of named entities in the extracted paragraphs (and increase the Lucene score in accordance with the number of identified named entities);
- if the question type is Definition, then answers having the definition form, as identified by our grammar [2] are preferred;
- the length of the sentence and the distance (in number of words) between the focus, the named entities and the keywords counts to a certain extend.

Lucene scores are also considered when these heuristics are not able to differentiate between candidate answers. After applying these criteria, all paragraphs are awarded a score and the paragraph with the biggest score is chosen as containing the correct answer. Example of applying a rule on several answer candidates is given below, when the name entities appear in the answer list:

Question: *At what age did Albert Einstein publish his famous Theory of Relativity?*
Answer 1: *Theory of Relativity, a theory that...*
Answer 2: *Suppose Einstein would rather...*
Answer 3: *... Albert Einstein, which, at only 22 years old,...*

Not only does Answer 3 has two named entities (while Answer 2 has only one), but it also has a numeric piece of data (22), which is automatically found by our number searcher and, since the answer type is a numerical data, its score is boosted.

3 Results

For the 2009 ResPubliQA track, our team submitted runs for two language pairs: English-English and Romanian-Romanian. The best runs results are shown in Table 1.

Table 1. Results of UAIC's best runs

	RO-RO	**EN-EN**
answered right	236	243
answered wrong	264	204
total answered	500	447
unanswered right	0	18
unanswered wrong	0	35
unanswered empty	0	0
total unanswered	0	53
c@1 measure	**0.47**	**0.54**

Each answer was evaluated as being *right* or *wrong*, and the unanswered questions were also allowed to be *empty*. The evaluation method and the track requirements were significantly different from those of past years, so a direct comparison between our previous results and this edition scores is difficult. However, the simplification of the answer extraction step, by requesting paragraphs as answers and not exact answers, did have a major impact in the improvement of our scores as, according to our estimates, this step accounted for about 24.8% of our previous errors.

For the two languages we sent runs, we tried two different decision approaches: for the *En-En* task, we considered a confidence level below which answers were marked *unanswered*. Choosing to ignore this unanswered feature for Romanian had a negative impact on our score (the results were 0.07 better for the *En-En* run).

The evaluation of the performance of our question-answering systems participating in the ResPubliQA competition concentrated on detecting the module that performed worst. Thus, we started by observing the question analysis module. Only the question

type identification was significantly modified as compared to edition from 2008[4] (see Section 2.2). The observations concerning the question type recognition module performance are presented in table 2.

Table 2. Evaluation of the question identification module

Gold /UAIC-Run	Factoid	Procedure	Purpose	Reason	Definition
Factoid	**82.73%**	12.95%	0.72%	0.72%	2.88%
Procedure	12.66%	**87.34%**	0.00%	0.00%	0.00%
Purpose	12.77%	1.06%	**37.23%**	17.02%	31.91%
Reason	3.23%	1.08%	1.08%	**93.55%**	1.08%
Definition	2.11%	2.11%	1.05%	0.00%	**94.74%**

Table 2 represents a confusion table for the five question types: the columns are the gold question types, and the rows correspond to the question types identified by our system. For instance, the first cell on the first raw corresponds to the percentage of *Factoid* questions that were identified as *Factoid* by our system (82.73%). The next cell on the same raw represents the percentage of *Factoid* questions that were considered *Procedure* by our system (12.95%), etc. The bold cells represent the percentage of correct identification for each type. The system identifies, with a good rate of success, most of the question types, except for the *Purpose* ones, which are identified, for the most part, with *Definition* questions.

It was straightforward to believe that an accepted answer to a question depends very much on correctly identifying the question type, it is not mandatory. Table 3 shows the number of questions correctly/incorrectly answered as compared to their type identification (the first two columns are questions correctly identified by the system, which have a correct, respectively an incorrect answer, and the last two columns correspond to the number questions incorrectly identified by the system).

Table 3. UAIC's best run for the RO-RO track

Question Type	Right identified Type (396)		Wrong Identified Type (104)	
Evaluation	Correctly Answered	Incorrectly Answered	Correctly Answered	Incorrectly Answered
Factoid	62	53	11	13
Definition	41	49	4	1
Reason	49	38	3	3
Procedure	24	45	4	6
Purpose	15	20	23	36

The results show that the system answered correctly for 38.2% of the correctly identified question types, indicating that the main weakness of our system should be looked elsewhere.

[4] Thus, we will only present the error rates of the question type identification module, directing the reader towards [1] for a detailed analysis of the other sub-modules of our question analysis module.

In order to determine the performance of the answer extraction procedure, we evaluated the number of questions that could be correctly answered, (the answer for which could be found in the returned snippets but was not selected). The results show that, out of the 264 questions with incorrect answers, only 35 had the correct answer in the ranked list returned by the retrieval module. For the rest of them, the correct paragraph was not in the candidate list. This huge difference (86.74% of incorrect answers are due to the lock of the good paragraph in the candidate list) indicates that the biggest problem of our system this year was the retrieval module. Since we have used Lucene for both indexing and retrieval, we determined that the flaw appeared at the query generation phase. For questions of type *Definition* and *Reason*, the queries were built using as much information as possible from the questions, but also some heuristics inspired by analysing the development set.

The answer extraction module was refined for the *Definition* type questions, using the Romanian grammar presented in [2], but also for *Reason* questions. The question type for which the module performed worst was the *Purpose* type, mainly due to the fact that the patterns extracted from the development questions were too sparse.

It is important to mention that the gold result file provided by the organizers has a small drawback when used to assess the system performance, because the 500 questions provided were built on the basis of the Acquis collections in different languages and for uniformity purposes, the gold file only contains the gold answer (document and paragraph id) in the original language and in English. For instance, the sentence "*¿Qué son los "bancos centrales nacionales?*" has as gold answer the document id and paragraph in Spanish (the original language of the question) and in English, with the translation of the answer together with the document and paragraph id of the English correspondence. When evaluating the performance of the Ro-Ro system, an alignment of the English documents and paragraphs indexes to Romanian was needed. We used the JRC-Acquis alignment available on the JRC site, created using the Vanilla[5] aligner. The problem was that, out of the 500 answer paragraphs, only 243 were found aligned to the Romanian version of the Acquis using the English-Romanian alignment; therefore, for more than half of the questions, we missed the "official" gold Romanian paragraphs in order to search for the candidate answer list for a precise evaluation of the retrieval module's accuracy. To overcome this drawback, we used the baseline file for the Ro-Ro task to compensate, in case we did not find the official alignment. For the cases in which we did not find an answer, we evaluated by checking on both documents (English and Romanian) and choosing the correct alignment.

4 Conclusions

This paper presents the Romanian Question Answering system which took part in the QA@CLEF 2009 competition. The evaluation shows an overall accuracy of 47% on RO-RO and 54% on EN-EN, which are still our best results from 2006 till now.

Two major improvements were made this year: first we continued to eliminate the most time-consuming modules from the pre-processing steps. Secondly, important

[5] Vanilla aligner: `http://nl.ijs.si/telri/Vanilla/`

improvements were made regarding the information retrieval module, where Lucene queries were built in a specific way for Definition and Reason questions. Also, we used a Romanian grammar in order to extract answers for definition questions.

Another reason for the good results obtained in this edition is the elimination of two important error sources. Firstly, the corpus used to extract the answers was the JRC-Acquis corpus in XML format which required no additional pre-processing (in the previous edition 10% of the errors were due to a wrong pre-processing of the corpus). Secondly, we didn't need to extract the exact answer from candidate paragraphs (last year, in 24.8% of the cases, we selected the incorrect answer from the correctly extracted paragraph).

We consider that the results of our question answering system can still be improved by making use of semantic information in the retrieval phase, as well as by improving the question type and answer extraction procedures in the case of *Purpose* and *Procedure* questions.

Acknowledgements

This paper presents the work of the Romanian team in the frame of the PNCDI II, SIR-RESDEC project number D1.1.0.0.0.7/18.09.2007.

References

1. Correa, S., Buscaldi, D., Rosso, P.: NLEL-MAAT at CLEF-ResPubliQA. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)
2. Iftene, A., Pistol, I., Trandabăţ, D.: UAIC Participation at QA@CLEF2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 385–392. Springer, Heidelberg (2009)
3. Iftene, A., Trandabăţ, D., Pistol, I.: Grammar-based Automatic Extraction of Definitions and Applications for Romanian. In: Proc. of RANLP Workshop "Natural Language Processing and Knowledge Representation for eLearning Environments, Borovets, Bulgaria, pp. 19–25, (September 26, 2007) ISBN 978-954-452-002-1
4. Ion, R., Ştefănescu, D., Ceauşu, A., Tufiş, D., Irimia, E., Barbu-Mititelu, V.A., Trainable Multi-factored, QA System. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, 30 September- October 2 (2009)
5. Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, Á., Forăscu, C., Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P.: Overview of ResPubliQA 2009. In: Question Answering Evaluation over European Legislation (2009)
6. Rodrigo, Á., Pérez, J., Peñas, A., Garrido, G., Araujo, L.: Approaching Question Answering by means of Paragraph Validation. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece, September 30- October 2 (2009)

Studying Syntactic Analysis in a QA System: FIDJI @ ResPubliQA'09

Xavier Tannier and Véronique Moriceau

LIMSI-CNRS, University Paris Sud 11, Orsay, France
xtannier@limsi.fr, moriceau@limsi.fr

Abstract. FIDJI is an open-domain question-answering system for French. The main goal is to validate answers by checking that all the information given in the question is retrieved in the supporting texts. This paper presents FIDJI's results at ResPubliQA 2009, as well as additional experiments bringing to light the role of linguistic modules in this particular campaign.

1 Introduction

This paper presents FIDJI's results in ResPubliQA 2009 [1] for French. In this task, systems receive 500 independent questions in natural language as input, and return one paragraph containing the answer from the document collection. FIDJI[1] (Finding In Documents Justifications and Inferences) is an open-domain question-answering system for French, which uses syntactic information, especially dependency relations. The goal is to match the dependency relations derived from the question and those of a passage and to validate the type of the potential answer in this passage or in another document.

Many QA systems use syntactic information, especially dependency relations, mainly for answer extraction. A basic approach consists in looking for an exact matching between the dependency relations of the question and those of the passage [2]. In [3], the dependency parse tree and the semantic structure of the question are used for answer extraction after a syntactic and semantic parsing. Some research is also dedicated to question decomposition for QA. In [4], a strategy for decomposing questions at a syntactic and semantic level is proposed: when the QA system START cannot find answers to a question, it tries to answer each sub-question. The system uses a number of parameterized annotations and semantic templates applied to the whole collection of documents in order to relate questions to information in one or several documents. FERRET [5], an interactive QA system, performs a syntactic and semantic decomposition of complex questions which aims at splitting a complex question into a set of semantically simpler questions that the system can answer easily. Finally, six decomposition classes (temporal, meronymy, etc.) are presented in [6] and are employed for annotating German questions and triggering different decomposition methods.

[1] This work has been partially financed by OSEO under the Quaero program.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 237–244, 2010.

Almost all recent researches are based on a syntactic and semantic analysis and often imply a pre-processing of the whole document collection. Our aim is to extract and validate answers by going beyond the exact syntactic matching between questions and answers, without using any semantic resources and with as less pre-processing as possible.

After a brief overview of the system, this paper presents the results obtained at the campaign ResPubliQA 2009, as well as some experiments bringing to light the role of linguistic modules in this particular campaign. We show notably that syntactic analysis, that proved useful in other campaigns, decreases results in this particular case.

2 FIDJI

When a piece of information is being searched, it can be formulated in different ways and some knowledge bases or inferences may be useful to identify it. But, even if lexical databases containing term variations exist (*e.g.* synonyms), conceptual databases for French are not available and, consequently, a semantic approach is not possible. Therefore, our approach consists in extracting and validating answers by using syntactic information, in particular syntactic dependency relations. The main goal is to validate answers by checking that all the features identified by the question analysis (see Section 2.1) are retrieved in the supporting texts.

Our answer validation approach assumes that the different entities of the question can be retrieved, properly connected, either in a sentence, in a passage or in multiple documents. We designed the system so that no particular linguistic-oriented pre-processing is needed. The document collection (JRC-Acquis about EU documentation) is indexed by the search engine Lucene[2] [7]. First, FIDJI submits the keywords of the question to Lucene: the first 100 documents are then processed (syntactic analysis and named entity tagging). Among these documents, the system looks for sentences containing the most syntactic relations of the question. Finally, answers are extracted from these sentences and the answer type, when specified in the question, is validated. Figure 1 presents the architecture of FIDJI and more details can be found in [8,9].

Next sections summarize the way FIDJI extracts answers and focus on ResPubliQA specificities.

2.1 Syntactic Analysis

FIDJI has to detect syntactic implications between questions and passages containing the answers. Our system relies on syntactic analysis provided by XIP, which is used to parse both the questions and the documents from which answers are extracted. XIP [10] is a robust parser for French and English which provides dependency relations and named entity recognition; some rules and features have also been added [9].

[2] http://lucene.apache.org/java/docs/index.html

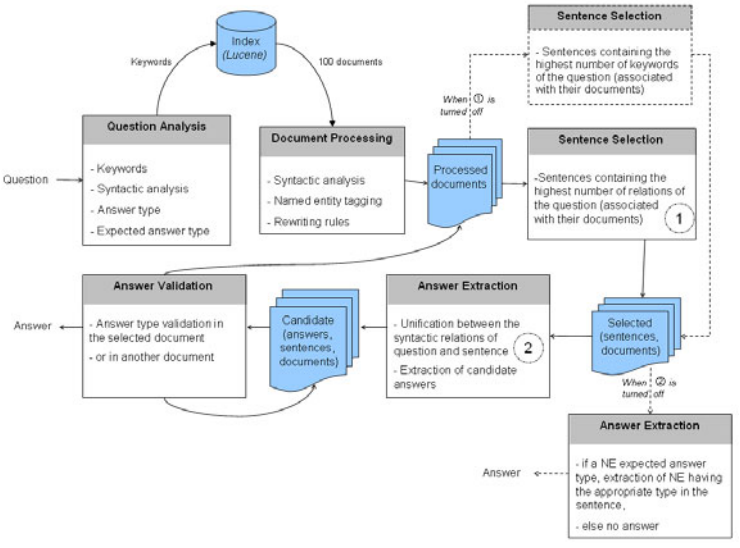


Fig. 1. Architecture of FIDJI

Question analysis consists in identifying:

- The syntactic dependencies given by XIP;
- The keywords submitted to Lucene (words tagged as noun, verb adjective or adverb by XIP);
- The question type:
  - Factoid (concerning a fact, typically who, when, where questions),
  - Definition (What is...),
  - Boolean (expecting a yes/no answer),
  - List (expecting an answer composed of a list of items),
  - Complex questions (why and how questions): *reason* and *purpose* questions in ResPubliQA are identified as why questions by FIDJI while *procedure* questions are identified as how questions.
- The expected type(s): NE type and/or (specific) answer type.

The answer to be extracted is represented by a variable (ANSWER) introduced in the dependency relations. The slot noted 'ANSWER' is expected to be instantiated by a word, argument of some dependencies in the parsed sentences. This word represents the answer to the question (see Section 2.2). The question type is mainly determined on the basis of the dependency relations given by the parser. Examples are given in the following sections.

2.2 Extracting Candidate Paragraphs

In ResPubliQA, answers are not focused, short parts of texts, but full paragraphs that must contain the answer. Passages are not indefinite parts of texts

of limited length, but predefined paragraphs identified in the corpus by XML tags <p>. FIDJI usually works at sentence level. For the aim of ResPubliQA specific rules, we chose to work at paragraph level. This consisted in specifying that sentence separators were <p> XML tags in the collection, rather than usual end-of-sentence markers.

Although answers to submit to the campaign are full paragraphs, our system is designed to hunt down short answers. For most questions, typically factoid questions, it is still relevant to find short answers, and then to return a paragraph containing the best answer. This is not the case of 'how' or 'why' questions, where no short answer may be retrieved.

Once documents are selected by the search engine and analyzed by the parser, FIDJI compares the document paragraphs with question analysis, in order to 1/ extract candidate answers or select a relevant paragraph, and 2/ give a score to each answer, so that final answers can be ranked.

Factoid Questions. Within selected documents, candidate paragraphs are those containing the most dependencies from the question. Once these paragraphs are selected, two cases can occur:

1. Question dependencies with an 'ANSWER' slot are found in the sentence. In this case, the lemma instantiating this slot is the head of the answer. The full answer is composed of the head and its basic modifiers. The eventual NE type and answer type of this answer are checked. Answer type can be validated by different syntactic relations in the text: definition (*"The French Prime minister, Pierre Bérégovoy"*), attributNN (*"Pierre Bérégovoy is the French Prime minister"*), etc.
2. The 'ANSWER' does not unify with any word of the passage. In this case, the elements having an appropriate NE type and/or answer type are selected.

If no possible short answer is found, the paragraph is still considered as a candidate answer. But in any case, a paragraph containing an extracted short answer will be prefered if it exists. For example:
187 - *Quel pays hors de l'Union peut exprimer son intention de participer à des opérations militaires ? (Which country outside the Union may express its intention of taking part in military operations?)*

- Syntactic dependencies and NE tagging:

```
DEEPSUBJ(pouvoir, ANSWER)          PREPOBJ(participer, à)
DEEPSUBJ(exprimer, ANSWER)         VMOD(participer, opération)
DEEPOBJ(pouvoir, exprimer)         ATTRIBUTADJ(opération, militaire)
DEEPOBJ(exprimer, intention)       LOCATION[COUNTRY](ANSWER)
```

- Question type: **factoid**
- Expected type: **location (country)**

The following passage is selected because it contains these dependencies:
(3) Si l'Union européenne décide d'entreprendre une opération militaire de gestion de crise en ayant recours aux moyens et capacités de l'OTAN, la République

de Turquie peut exprimer son intention de principe de participer à l'opération. ((3) If the European Union decides to undertake a military crisis management operation with recourse to NATO assets and capabilities, the Republic of Turkey may express its intention in principle of taking part in the operation.)

```
DEEPSUBJ(décider, union européen)          CONNECT(décider, si)
DEEPSUBJ(entreprendre, union européen)
PREPOBJ(entreprendre, de)                  DEEPOBJ(entreprendre, opération)
ATTRIBUTADJ(opération, militaire)          ATTRIBUT_DE(opération, gestion)
...                                        ...
```

DEEPSUBJ(pouvoir, république de turquie)
DEEPSUBJ(exprimer, république de turquie)

```
DEEPOBJ(pouvoir, exprimer)                 DEEPOBJ(exprimer, intention)
ATTRIBUT_DE(intention, principe)           PREPOBJ(participer, de)
VMOD(participer, opération)                PREPOBJ(participer, à)
ORG(otan)                                  ORG(union européen)
```

LOCATION[COUNTRY](république de turquie)

The slot 'ANSWER' is instantiated by *République de Turquie*. Finally, the expected answer type is validated as the selected answer is tagged as a country.

Complex Questions. Complex questions ('how', 'why', etc.) do not expect any short answer. On these kinds of questions, the system behaves more as a passage retrieval system. The paragraphs containing the more syntactic dependencies in common with the question are selected. Among them, the best-ranked is the one that is returned first by Lucene.

2.3 Scoring

FIDJI's scores are not composed of a single value, but of a list of different values and flags. The criteria are listed below, in decreasing order of importance:

- A paragraph containing an extracted short answer will be prefered if it exists.
- NE value (appropriate NE value or not – only for factoid questions).
- Keyword rate (between 0 and 1, the rate of question major keywords present in the passage: proper names, answer type and numbers).
- Answer type value (appropriate answer type or not – only for factoid questions).
- Frequency weighting (number of extracted occurrences of this answer – only for factoid questions).
- Document ranking (best rank of a document containing the answer, as returned by the search engine. If this case, the lower the better).

3 Results

Table 1 presents FIDJI's results at ResPubliQA by types of questions (note that we found 26 questions that were ill-formed or with misspellings). Only

one answer per question was allowed, so the values correspond to the rate of correct answers for each question type. With an overall accuracy of 0.3, FIDJI ranked 2nd out of 3 participants for French but only 15th out of 20 for all languages. Unfortunately, details concerning the best ranked system have not been published so far. The 3rd-ranked system uses similar techniques as FIDJI, but also semantic parsing [1].

Results are lower than former campaigns' scores, especially concerning factoid and definition questions. All participants for French had also lower results than those of the "pure information retrieval" baseline [1] which consisted in querying the indexed collection with the exact text of the question and returning the paragraph retrieved in the first position.

In a former study [11], we showed that modules using syntactic analysis (modules ① and ② in Figure 1) improved significantly the results in comparison with a traditional keyword search. These experiments had been conducted on former CLEF collections (newspapers *Le Monde* and *ATS*) as well as *Quaero* web corpus. We ran our system on ResPubliQA collection with both modules switched off (see Table 2). Passage extraction is then performed by a classical selection of sentences containing a maximum of question significant keywords, and answer extraction is achieved without slot instantiation within dependencies.

This unofficial run leads to 191 correct answers and 35 "NOA" ($c@1 = 0.42$). This is surprisingly much higher than our official run ($c@1 = 0.30$), but confirms the comparable results obtained by the baselines ($c@1 = 0.45$), as well as lower results obtained by the 3rd-ranked system ($c@1 = 0.23$), that uses also syntactic analysis. As we can see in Tables 1 and 2, the identification of question type by FIDJI is good and results are better for every type of questions when syntactic modules are switched off. So, neither the performance of the question analysis modules nor the question types can explain the lower results. This interesting issue deserves a specific analysis. Our linguistic processing is useful in general (CLEF, Quaero) but harmful in ResPubliQA, with the following specificities:

- Specific guidelines: the final answer is an entire paragraph instead of a focused short answer. The answer extraction module becomes then naturally less useful.
- Specific domain and questions: different register of language, more constrained vocabulary and very frozen and unusual way to express ideas. This

Table 1. FIDJI results by question types

Question type	**Number of questions**	**Correct identification of question type**	**Correct answer**
Factoid	116	88.8%	36.2 %
Definition	101	91%	15.8 %
List	37	91.9%	16.2 %
Procedure	76	86%	22.4 %
Reason/Purpose	170	97%	40 %
TOTAL	500	93.6%	30.4 %

Table 2. Comparison of results with and without syntactic modules

	Factoid	Definition	List	Procedure	Reason	TOTAL
With syntactic modules	36.2%	15.8%	16.2%	22.4%	40%	30.4%
Without syntactic modules	41.4%	9.5%	34.3%	40.5%	50.8%	38.2%

is particularly true for definitions, quite easy to detect in newspaper corpora, that have been poorly recognized for this evaluation (see example 2 below).
- Specific documents: texts in the document collection have a very particular structure, with an introduction followed by long sentences extending on several paragraphs and having all the same skeleton (*e.g. Having regard to. . . , Whereas . . . For the purpose of this Directive. . .*).

Looking carefully at the results shows that, in these particular documents, using syntactic dependencies as the main clue to choose paragraph candidates is not always a good way to find out a relevant passage. This is especially true for complex questions, but not only. Indeed, the selection of the paragraph containing the most question dependencies often leads to the introduction of the document or to a very general paragraph containing poor information.

Example 1: 0006 - *What is the scope of the council directive on the trading of fodder seeds?* is answered by:
COUNCIL DIRECTIVE of 14 June 1966 on the marketing of fodder plant seed (66/401/EEC)
containing many dependencies but answering nothing, while a good result was later in the same document, but with an anaphora:
This Directive shall apply to fodder plant seed marketed within the Community, irrespective of the use for which the seed as grown is intended.
Dependency relations are still useful to find a good document, but often fails to point out to the correct paragraph.

Example 2: 0068 - *What is the definition of the term "Operation TIR"?* is not answered correctly. In English, the answer provided by the gold standard is:
For the purposes of this Convention: (a) the term "TIR operation" shall mean the transport of goods
In the French collection, this paragraph is split:
* Aux fins de la présente convention , on entend : *
* a) par "opération TIR", le transport de marchandises ... *
that can be translated into:
* For the purposes of this Convention, we mean: *
* a) by "operation TIR", transport of goods ... *
First, this is a quite unusual way to introduce a definition. Second, in the French document, the sentence is split into two paragraphs. In addition to the difficulty of identifying a definition pattern over several paragraphs and extracting the answer in this case, this raises the problem of cross-language comparison of systems, since corpora are not exactly parallel.

4 Conclusion

We presented in this paper our participation to the campaign ResPubliQA 2009 in French. The aim of this campaign was to study "if current QA technologies tuned for newswire collections can be easily adapted to a new domain (law domain in this case)". In our particular case, we adapted our syntactic-based QA system FIDJI in order to produce a single long answer in the form of JRC-Acquis tagged paragraphs. The system got much lower results than usual, and this variation can be explained by many particularities of this campaign: new domain, different register of language, different structure of documents and different guidelines.

Different experiments on the collection confirmed that the use of syntactic analysis decreased results, whereas it proved to help when used in other campaigns. This shows that syntactic analysis should be used in different manners according to the type of tasks and documents.

References

1. Peñas, A.P., Forner, P., Sutcliffe, R., Rodrigo, A., Forăscu, C., Iñaki Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P.: Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation.In: [12] (2009)
2. Katz, B., Lin, J.: Selectively Using Relations to Improve Precision in Question Answering. In: Proceedings of the EACL 2003 Workshop on Natural Language Processing for Question Answering (2003)
3. Sun, R., Jiang, J., Tan, Y.F., Cui, H., Chua, T.S., Kan, M.Y.: Using Syntactic and Semantic Relation Analysis in Question Answering. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Fourteenth Text REtrieval Conference (2005)
4. Katz, B., Borchardt, G., Felshin, S.: Syntactic and Semantic decomposition Strategies for Question Answering from Multiple Resources. In: Proceedings of the AAAI 2005 Workshop on Inference for Textual Question Answering (2005)
5. Hickl, A., Wang, P., Lehmann, J., Harabagiu, S.: FERRET: Interactive Question-Answering for Real-World Environments. In: COLING/ACL Interactive Presentation Sessions, Sydney, Australia (2006)
6. Hartrumpf, S., Glöckner, I., Leveling, J.: University of Hagen at QA@CLEF 2008: Efficient Question Answering with Question Decomposition and Multiple Answer Streams. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, Springer, Heidelberg (2009)
7. Hatcher, E., Gospodnetić, O.: Lucene in Action. Manning (2004)
8. Moriceau, V., Tannier, X., Grau, B.: Utilisation de la syntaxe pour valider les réponses à des questions par plusieurs documents. In: Proceedings of COnférence en Recherche d'Information et Applications, CORIA (2009)
9. Tannier, X., Moriceau, V.: FIDJI in ResPubliQA 2009. In: [12] (2009)
10. Aït-Mokhtar, S., Chanod, J.P., Roux, C.: Robustness beyond Shallowness: Incremental Deep Parsing. Natural Language Engineering 8, 121–144 (2002)
11. Moriceau, V., Tannier, X.: Apport de la syntaxe dans un sysème de question-réponse: étude du système FIDJI. In: Actes de la Conférence Traitement Automatique des Langues Naturelles, Senlis, France (2009)
12. Working Notes for the CLEF 2009 Workshop, Corfu, Greece (2009)

Approaching Question Answering by Means of Paragraph Validation

Álvaro Rodrigo, Joaquín Pérez-Iglesias, Anselmo Peñas,
Guillermo Garrido, and Lourdes Araujo

NLP & IR Group, UNED, Madrid
{alvarory,joaquin.perez,anselmo,ggarrido,lurdes}@lsi.uned.es

Abstract. In this paper we describe the QA system developed for taking part in Res-PubliQA 2009. Our system was composed by an IR phase focused on improving QA results, a validation step for removing paragraphs that are not promising and a module based on ngrams overlapping for selecting the final answer. Furthermore, a selection module that uses lexical entailment in combination with ngrams overlapping was developed in English. The IR module achieved very promising results that were improved by the ngram ranking. Moreover, the ranking was slightly improved when lexical entailment was used.

1 Introduction

The first participation of UNED at QA@CLEF (called this year ResPubliQA) is based on our experience as participants and organizers of the Answer Validation Exercise[1] (AVE) [4,5,8,9]. Our motivation for using Answer Validation (AV) in this task comes from the conclusions obtained in AVE, where it was shown that AV systems could contribute towards the improvement of results in Question Answering (QA).

Besides, the evaluation in this edition gives a higher reward for not giving an answer than for returning an incorrect one, what suggests the use of AV. Thus, if our QA system considers that there is no correct answer among the candidate ones to a question, then no answer is returned to that question.

In this paper we describe the main features of our QA system and the results obtained in monolingual English and Spanish. The rest of this paper is structured as follows: In Section 2 we describe the main components of the system. The description of the submitted runs is given in Section 3, while the results and their analysis are shown in Section 4. Finally, some conclusions and future work are given in Section 5.

2 System Overview

The main steps performed by our system are described in detail in the following subsections.

[1] http://nlp.uned.es/clef-qa/ave

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 245–252, 2010.

2.1 Retrieval Phase

A first selection of paragraphs that are considered relevant for the proposed questions is performed in this phase. The goal is to obtain a first set of paragraphs (no more than 100 per question) ordered according to their relevance to the question. We used BM25 [7], which can be adapted to fit the specific characteristics of the data in use. More information about the selected retrieval model can be found in [6], where the retrieval model and its successful results are described in more detail.

2.2 Pre-processing

Each question and each paragraph returned by the retrieval phase is pre-processed in this step with the purpose of obtaining the following data:

- **Named Entities (NEs):** the FreeLing NE recognizer [1] is applied in order to tag proper nouns, numeric expressions and temporal expressions of each question and each candidate paragraph. Besides, information about the type of the proper noun is included. That is, for proper nouns we have types PERSON, ORGANIZATION and LOCATION[2].
- **Lemmatization:** the FreeLing PoS tagger in Spanish and TreeTagger[3] in English are used for obtaining the lemmas of both paragraphs and questions.

2.3 Paragraph Validation

The objective of this step is to remove paragraphs that do not satisfy a set of constraints imposed by a question since, in that case, it is not likely to find a correct answer for that question in these paragraphs. A set of modules for checking constraints have been implemented (3 in this edition) and they are applied in a pipeline processing. That is, only paragraphs able to satisfy a certain constraint are checked against the following constraint. In fact, it is possible to obtain no paragraph as output, what means that no paragraph is a candidate for containing a correct answer. The constraints implemented in this edition are explained in the following subsections.

Expected Answer Type Matching. Only paragraphs that contain elements of the expected answer type are validated in this module. Firstly, the expected answer type is detected for each question. We based our taxonomy on the one used in the last editions of QA@CLEF. Thus, we considered the following answer types: *count*, *time*, *location*, *organization*, *person*, *definition* and *other*.

For performing the matching process we took advantage of the fact that all the types in our taxonomy (except *definition* and *other*) match the possible NE types tagged in the pre-processing step. That is, *count* questions must be answered by numeric expressions, *time* questions must be answered by temporal expressions, etc. Then, the module validates paragraphs that contain at least a NE of the expected answer type and rejects

[2] This information is given by Freeling only for Spanish texts. A generic *proper noun* type is used in English texts.

[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

the other paragraphs. In case of the expected answer type is *definition* or *other*, all the input paragraphs are validated because the system does not have enough evidences for rejecting them.

Our system can perform two kinds of expected answer type matching: the coarse grained matching and the fine grained matching. In the fine grained matching all the possible expected answer types and all the possible NE types are used, while in the coarse grained matching some types are grouped. Due to space restrictions, more details about this module can be found in [10].

NE Entailment. The validation process performed by this module follows the intuition that the NEs of a question are such important pieces of information that they must appear in a correct answer [8].

This module receives as input the NEs of a question and their candidate paragraphs. Then, only paragraphs that contain all the NEs of the question are validated. If a question does not have any NE, all the paragraphs are validated because there are no evidences for rejecting them.

Acronym Checking. This module works only over questions that ask about the meaning of a certain acronym, as for example *What is NATO?* or *What does NATO stand for?* The objective is to validate only paragraphs that could contain an explanation for these acronyms. If the restriction cannot be applied, all the input paragraphs are validated.

Firstly, the module checks whether the question is of *definition* type and whether it is asking about a word that only contains capitalized letters, which we called acronym. If the question satisfies these constraints, the acronym is extracted.

Secondly, only paragraphs that can contain a possible definition for the extracted acronym are validated. In the current implementation it is considered that if a paragraph contains the acronym inside a pair of brackets, then it might contain a definition of the acronym and the paragraph is validated.

2.4 Paragraph Selection

After some experiments performed at the development period, we based the decision of which paragraph to select on the overlapping between questions and answer paragraphs. The paragraph selection module works only when the validation process returns more than one candidate paragraph. If there is only one candidate paragraph, then it is the one selected. If there is no candidate paragraph, that means that no candidate paragraph was suitable for containing a correct answer. In these cases the system does not answer the question and the paragraph that was chosen by the IR engine at the first position is the one returned as the hypothetical answer.

We have two modules for selecting the final answer: one based only on lemmas overlapping and another one based on lemmas overlapping and Lexical Entailment. Both modules are described below.

Setting 1. We discarded stop words and measured overlapping using lemmas as a way of avoiding different formulations of similar expressions. Thus, the selection process works as follows:

1. Overlapping using 1-grams (lemmas) is measured. If the maximum overlapping with the question is achieved for only one paragraph, then that paragraph is selected. If the maximum overlapping is achieved for more than one paragraph, the next step is performed.
2. The overlapping using 2-grams (lemmas) is measured over the paragraphs with the maximum overlapping using 1-grams. If the maximum overlapping with the question is achieved for only one paragraph, then that paragraph is selected. If the maximum overlapping is achieved for more than one paragraph, the process is repeated with 3-grams, 4-grams and 5-grams stopping when there is still more than one paragraph with the maximum overlapping using 5-grams (lemmas) to perform the next step.
3. If there is more than one paragraph with the maximum overlapping using 5-grams (lemmas), then the one which obtained the higher ranking in the IR process is selected.

Setting 2. We developed in English another version for the selection process that is based on Lexical Entailment. For this purpose we took advantage of a module, which works only in English, based on WordNet relations and paths for checking the entailment between lexical units [2]. The same process performed in setting 1 is applied, but there can be overlapping between a word in a paragraph and a word in a question if the two words are the same or the word in the paragraph entails (according to the entailment module based on WordNet) the word in the question.

3 Runs Submitted

We took part in two monolingual tasks (English and Spanish), sending two runs for each of these tasks. All the runs applied the same IR process and the main differences are found in the validation and selection steps. The characteristics of each run were as follows:

- **Monolingual English runs:** both runs applied for the validation process the coarse grained expected answer type matching (because the NE recognizer allowed us to use only this kind of matching), the NE entailment module and the acronym checking module. The differences come in the paragraph selection process:
  - **Run 1:** paragraph selection was performed by the module based on lemmas overlapping (setting 1) described in Section 2.4.
  - **Run 2:** paragraph selection was performed by the module based on lemmas overlapping and Lexical Entailment (setting 2) described in Section 2.4. The motivation for using this selection module was to study the effect of Lexical Entailment for ranking answer paragraphs.
- **Monolingual Spanish runs:** in both runs the selection process was based on lemmas overlapping (setting 1 described in Section 2.4). Both runs applied the validation step in the same way for both the NE entailment module and the acronym checking module. The differences come in the use of the expected answer type matching module:

- **Run 1:** the fine grained expected answer type matching was applied.
- **Run 2:** it was applied the coarse grained expected answer type matching. The objective was to study the influence of using a fine grained or a coarse grained matching. It may be thought that the best option is the fine grained matching. However, possible errors in the classification given by the NE recognizer could contribute to obtain better results using the coarse grained option.

4 Analysis of the Results

The runs submitted to ResPubliQA 2009 were evaluated by human assessors who tagged each answer as *correct* (R) or *incorrect* (W). In order to evaluate the performance of systems validating answers, the task allowed to return an hypothetical candidate answer when it was chosen not to answer a question. These answers were evaluated as *unanswered* with a *correct* candidate answer (UR), or *unanswered* with an *incorrect* candidate answer (UI). The main measure used for evaluation was *c@1*, while *accuracy* was used as a secondary measure[4].

The results obtained for the runs described in Section 3 are shown in Table 1 for English and Table 2 for Spanish. The results of a baseline system based only on the IR process described in Section 2.1 appear also in each Table. The answer given to each question in this baseline was the first one according to the IR ranking.

Table 1. Results for English runs

Run	#R	#W	#UR	#UI	accuracy	c@1
run 2	288	184	15	13	0.61	0.61
run 1	282	190	15	13	0.59	0.6
baseline	263	236	0	1	0.53	0.53

Table 2. Results for Spanish runs

Run	#R	#W	#UR	#UI	accuracy	c@1
run 1	195	275	13	17	0.42	0.41
run 2	195	277	12	16	0.41	0.41
baseline	199	301	0	0	0.4	0.4

4.1 Results in English

Regarding English results, run 2 achieves a slightly higher amount of correct answers than run 1 that is not significant. Since the only difference between both runs was the fact that run 2 used Lexical Entailment for ranking the candidate answers, the improvement was a consequence of using entailment. Although this is not a significant result for showing the utility of using entailment for ranking results in QA, it encourages us to explore more complex ways of using entailment for ranking paragraphs.

[4] The formulation of both measures can be seen in [3].

Comparing English runs with the English baseline it can be seen how the results of the submitted runs are about 10% better according to the given evaluation measures. A preliminary study showed us that most of this variation in the results was a consequence of the different ways for ranking paragraphs and not of the inclusion of the validation step. Then, the lemmas overlapping ranking used for the selection of paragraphs has shown to be more appropriate for this task than the one based only on IR ranking when the QA system is working in English. Therefore, results suggest that it is useful to include information based on lemmas when ranking the candidate paragraphs.

4.2 Results in Spanish

The results of the Spanish runs are quite similar as it can be seen in Table 2. Actually, the differences are not significant. Since the only difference between both runs was the expected answer type matching performed, results suggest that there are no big differences between the option of using one or another expected answer type matching. We detected that some of the errors obtained when the fine grained expected answer type matching was applied were caused by errors in the NE classification given by the NE recognizer. The possibility of having these errors was one of the motivations for using also coarse grained matching. However, when there was this kind of errors with the fine grained matching, the coarse grained matching did not help to find a right answer. Then, the analysis of the results shows that the fine grained matching could contribute towards improving results, but it depends too much on the classification given by the NE recognizer.

On the other hand, if we compare both runs with the baseline run, we can see that the results according to the two evaluation measures are quite similar. This is different to the results obtained in English, where the submitted runs performed better than the baseline. This means that the lemmas overlapping used for the selection process worked better in English than in Spanish.

4.3 Analysis of Validation

Given that one of our objectives for taking part at ResPubliQA was to study the impact of validation, we have analyzed the contribution of the validation modules in our QA system. Despite the fact that the basic ideas of the modules where the same in both languages and the question set was also the same (the same questions but translated to each language), the number of questions where each of the validation modules was applied differ between languages. This was a consequence of different question formulations for each language and little variations in the implementation of modules for different languages (due to the particularities of each language). However, the number of questions that were left unanswered was almost the same in both languages as it can be seen in Tables 1 and 2.

Since the candidate answers given to unanswered questions were also evaluated, the precision of systems validating answers (proportion of unanswered questions where the hypothetical answer was incorrect) can be measured. Table 3 shows the validation precision of the submitted runs for English and Spanish. In each language, the validation precision obtained was the same for both runs.

Table 3. Validation precision of the submitted runs in English and Spanish

Language	Val. precision
English	0.46
Spanish	0.57

As it can be seen in Table 3, the validation precision is close to 50% (slightly higher in Spanish and slightly lower in English). Therefore, the validation process applied by our QA system has not behaved very well.

We studied the errors produced by the validation process and we found that most of the errors were produced by the NE entailment module. On one hand, the constraint of having all the NEs of the question into the answer paragraph seemed to be very restrictive because a paragraph sometimes can omit some NEs that have been referred before in the document. Therefore, in the future we would like to study a way of relaxing this constraint in order to improve results.

On the other hand, we found in Spanish some errors due to incorrect translations of the questions from English. For example, the NE *EEC* (which means European Economic Community) in question 17[5] was kept as *EEC* in Spanish, but the correct translation is *CEE* (which means *Comunidad Económica Europea*). This kind of errors in the translations caused that our system denied paragraphs that could contain correct answers.

Regarding the acronym checking, we found that its behaviour was quite good in Spanish but not in English. In fact, some questions were left unanswered in English because the acronym module was incorrectly applied. Therefore, we have to improve this module in English.

Finally, the expected answer type matching was applied in a low amount of questions for both languages and we did not observe too many problems in its performance. Now, we want to focus in improving its coverage so that it can be applied to a higher amount of questions.

5 Conclusions and Future Work

In this paper we have described our QA system and the results obtained for both English and Spanish monolingual tasks at ResPubliQA. The main steps of our system were an IR phase focused on improving QA results, a validation step for rejecting no promising paragraphs and a selection of the final answer based on ngrams overlapping.

The IR ranking has provided a good performance obtaining better results in English than in Spanish, while the validation process was not very helpful. On the other hand, the ranking based on ngrams was able to improve results of the IR module in English, while it maintains the performance in Spanish. Besides, Lexical Entailment has shown to be informative for creating the ranking of answers in English.

Future work is focused on solving the errors detected in each module, as well as developing validation modules for a broader range of questions. Furthermore, we want

[5] Why is it necessary to provide for information about certain foodstuffs in addition to those in Directive 79/112/EEC?

to perform a deeper study about the ranking of answers using ngrams in combination with Lexical Entailment and the information given by the modules used in the paragraph validation step.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation within the project QEAVis-Catiex (TIN2007-67581-C02-01), the TrebleCLEF Coordination Action, within FP7 of the European Commission, Theme ICT-1-4-1 Digital Libraries and Technology Enhanced Learning (Contract 215231), the Regional Government of Madrid under the Research Network MAVIR (S-0505/TIC-0267), the Education Council of the Regional Government of Madrid and the European Social Fund.

References

1. Carreras, X., Chao, I., Padró, L., Padró, M.: FreeLing: An Open-Source Suite of Language Analyzers. In: Proceedings of LREC 2004 (2004)
2. Herrera, J., Peñas, A., Rodrigo, Á., Verdejo, F.: UNED at PASCAL RTE-2 Challenge. In: Proceedings of the Second PASCAL Recognizing Textual Entailment Workshop (April 2006)
3. Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, Á., Forascu, C., Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P.: Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 174–196. Springer, Heidelberg (2010)
4. Peñas, A., Rodrigo, Á., Sama, V., Verdejo, F.: Overview of the Answer Validation Exercise 2006. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 257–264. Springer, Heidelberg (2007)
5. Peñas, A., Rodrigo, Á., Verdejo, F.: Overview of the Answer Validation Exercise 2007. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 237–248. Springer, Heidelberg (2008)
6. Pérez-Iglesias, J., Garrido, G., Rodrigo, Á., Araujo, L., Peñas, A.: Information Retrieval Baselines for the ResPubliQA Task. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 253–256. Springer, Heidelberg (2010)
7. Robertson, S., Walker, S.: Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In: Bruce Croft, W., van Rijsbergen, C.J. (eds.) SIGIR, pp. 232–241. ACM/Springer, New york (1994)
8. Rodrigo, Á., Peñas, A., Herrera, J., Verdejo, F.: The Effect of Entity Recognition on Answer Validation. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 483–489. Springer, Heidelberg (2008)
9. Rodrigo, Á., Peñas, A., Verdejo, F.: Overview of the Answer Validation Exercise 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 296–313. Springer, Heidelberg (2009)
10. Rodrigo, Á., Pérez, J., Peñas, A., Garrido, G., Araujo, L.: Approaching Question Answering by means of Paragraph Validation. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece (2009)

Information Retrieval Baselines for the ResPubliQA Task*

Joaquín Pérez-Iglesias, Guillermo Garrido, Álvaro Rodrigo, Lourdes Araujo, and Anselmo Peñas

NLP & IR Group, UNED, Madrid
{joaquin.perez,ggarrido,alvarory,lurdes,anselmo}@lsi.uned.es

Abstract. The baselines proposed for the ResPubliQA 2009 task are described in this paper. The main aim for designing these baselines was to test the performance of a pure Information Retrieval approach on this task. Two baselines were run for each of the eight languages of the task. Both baselines used the Okapi-BM25 ranking function, with and without a stemming. In this paper we extend the previous baselines comparing the BM25 model with Vector Space Model performance on this task. The results prove that BM25 outperforms VSM for all cases.

1 Overview

This year's ResPubliQA proposed the challenge of returning a right passage, containing a correct answer to a question, from a collection of more than a million paragraphs per language. The supplied collection was based on JRC-Acquis[1], a collection of EU documents. Both questions and documents are translated into different EU languages.

Our aim is twofold: to check what results can be obtained with a system based on pure IR techniques, and to establish a starting point for other participants in the task.

Passage retrieval has a well founded tradition that spans decades (see, for instance [5,1]). Different techniques can be found in the previous works specifically focused on applying a typical retrieval model to the selection of paragraphs or snippets within a document. In general, the techniques are adapted to the data characteristics beyond the straight use of the model.

A baseline like the ones proposed here can be considered a first phase within a typical pipeline architecture for Question Answering (QA). That is, a set of paragraphs considered relevant for the question are selected. The precision in terms

* This work has been partially supported by the Spanish Ministry of Science and Innovation within the project QEAVis-Catiex (TIN2007-67581-C02-01), the Treble-CLEF Coordination Action, within FP7 of the European Commission, Theme ICT-1-4-1 Digital Libraries and Technology Enhanced Learning (Contract 215231), the Regional Government of Madrid under the Research Network MAVIR (S-0505/TIC-0267), the Education Council of the Regional Government of Madrid and the European Social Fund.

[1] http://langtech.jrc.it/JRC-Acquis.html

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 253–256, 2010.

of retrieving the correct answer for the question within the top k paragraphs delimits in some manner the overall quality of the full QA system. In order to retrieve the most relevant paragraphs, the full collection has been indexed by paragraphs, removing a list of stopwords, and applying simple stemming algorithms for each language.

2 Retrieval Model

The selection of an adequate retrieval model is a key part of the task. By applying an inadequate retrieval function, a subset of candidate paragraphs where the answer cannot appear would be returned, and thus any subsequent technique applied to detect the answer within this subset will fail. In order to check the ability to retrieve the right paragraph by a pure information retrieval approach, two baselines were proposed. Both baselines are based on the Okapi-BM25 [4] ranking function, one with stemming as a pre-processing step and the other one without it.

In general, retrieval models are built around three basic statistics from the data: frequency of terms in a document; frequency of a term in the collection, where document frequency (DF) or collection frequency (CF) can be used; and document length. The ideal ranking function for this task should be adaptable enough to fit the specific characteristics of the data. For the ResPubliQA task, documents are actually paragraphs with an average length of ten terms, and the frequency of question terms within a paragraph hardly exceeds one. A good candidate paragraph for containing the answer of a question is one that has the maximum number of question terms (excluding stopwords) and has a length similar to the average (to avoid giving too much importance to term frequency within the paragraph).

The use of the classic Vector Space Model (VSM) [6] is not an adequate option for this task because this model typically normalises the weight assigned to a document with the document length. This causes that those paragraphs that contain at least one question term and have the lowest length will obtain the highest score. Moreover, the typical saturation of terms frequency used in this model, applying logarithm or root square, gives too much relevance to the term's frequency.

A more adequate ranking function for this task is BM25 [4]. In this ranking function, the effect of term frequency and document length on the final score of a document can be specified by setting up two parameters: b and k_1. We explain further the effect of these parameters over the ResPubliQA data in the following. The normalisation factor B which depends on the parameter b is computed as:

$$B = (1 - b) + b(\frac{dl}{avdl})$$

where dl is the document length, $avdl$ is the average document length and $b \in [0, 1]$. Assigning 0 to b is equivalent to avoiding the process of normalisation and therefore, the document length will not affect the final score ($B = 1$). If b is 1, we are carrying out a full normalisation $B = \frac{dl}{avdl}$.

Once the normalisation factor has been calculated, it is applied to term frequency. Final score is computed applying a term frequency saturation that uses the parameter k_1 allowing us to control the effect of frequency in the final score:

$$tf = \frac{freq_{t,d}}{B}; \quad idf_t = \frac{N - df_t + 0.5}{df_t + 0.5}; \quad R(q,d) = \frac{tf}{tf + k_1} \cdot idf_t$$

where: $\infty > k_1 > 0$; N is the total number of documents in the collection; $freq_{t,d}$ is the frequency of t in d and df_t is the document frequency of t. An implementation of the BM25 ranking function for Lucene was developed for this work[2]. The details of this implementation can be seen in [3]. The final expression for BM25 ranking function can be expresed as next:

$$R(q,d) = \sum_{t\ en\ q} \frac{freq_{t,d}}{k_1((1-b) + b \cdot \frac{dl}{avdl}) + freq_{t,d}} \cdot idf_t$$

3 Results and Conclusions

In order to test the precision of our retrieval system we proposed the execution of two baselines for each language. The paragraph selected in order to answer the question is the one ranked first in the retrieval phase. For the first baseline, a stemming process was applied[3], except for Bulgarian where no stemmer was available. The second baseline was built identically, except that no stemming was done. The parameter k_1 was fixed to a value of 0.1, after a training phase with the gold standard for the English development set supplied by the organisation. Its effect is reducing the influence of term frequency over the final score. We experimented with various settings for the b parameter.

The results of the baselines run were shown in [2], with a comparison over the different languages, this information is not showed here because of the lack of space. Here we will focus on one language, English, and compare with more detail the different performances when using VSM or BM25.

We compared the quality of the results for different values of the parameter b. The best results were obtained for $b = 0.4$, and the worst for $b = 1$ (as it can be expected). The average results for different values of b have been included in Table 1. The Table contains also information about whether a correct answer was found in the 5, 10 or 100 first retrieved paragraphs. For all cases, BM25 outperforms VSM, and the results obtained for different values of b are quite steady.

Some preliminary conclusions can be drawn from the results obtained, where BM25 outperforms VSM. Even the worst selection of the b parameter, both with and without stemming, yields better results than the VSM run. The obtained results with BM25 show a strong stability, since these results are only slightly affected by the b parameter. There is still a window for improvement optimising

[2] http://nlp.uned.es/~jperezi/Lucene-BM25/

[3] We used the Snowball implementation: http://snowball.tartarus.org/

Table 1. Comparison of BM25 against VSM. At the top, the results with stemming are shown. At the bottom, the results without stemming. We show the performance of BM25 for the best (0.4) and worst (1) selections of b, and the average from 0 to 1 using 0.1 as step. $found@k$ means whether a correct answer paragraph was found in the k first paragraphs returned by the IR ranking. Note that $P@1 = found@1$. In brackets the number of right paragraphs found.

Stemming

	P@1	**found@5**	**found@10**	**found@100**
BM25(b=0.4)	.56 (278)	.69 (347)	.74 (369)	.82 (412)
BM25(b=1)	.49 (246)	.67 (334)	.72 (358)	.82 (409)
BM25(average)	.53 (265.3)	.68 (342.7)	.73 (366.6)	.82 (411.6)
VSM	.42 (212)	.64 (321)	.70 (351)	.81 (406)

No Stemming

	P@1	**found@5**	**found@10**	**found@100**
BM25(b=0.4)	.55 (273)	.71 (356)	.75 (375)	.82 (408)
BM25(b=1)	.49 (246)	.67 (336)	.72 (358)	.81 (406)
BM25(average)	.53 (265.6)	.7 (348.7)	.74 (369.2)	.82 (407.6)
VSM	.43 (213)	.65 (323)	.71 (354)	.81 (403)

the k_1 parameter simultaneously with b. In relation with the stemming step, a clear similarity between runs with or without stemming can be observed. If we compare the average results of the BM25 runs with and without stemming process, we will see that no stemming outperforms stemming when precision is preferred over recall (for $P@1$, $found@5$, $found@10$), and it starts to be worse in situations where recall is preferable ($found@100$).

References

1. Hersh, W.R., Cohen, A.M., Roberts, P.M., Rekapalli, H.K.: TREC 2006 Genomics Track Overview. In: Voorhees, E.M., Buckland, L.P. (eds.) TREC, Volume Special Publication 500-272. NIST (2006)
2. Pérez, J., Garrido, G., Rodrigo, Á., Araujo, L., Peñas, A.: Information Retrieval Baselines for the ResPubliQA Task. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece (2009)
3. Pérez-Iglesias, J., Pérez-Agüera, J.R., Fresno, V., Feinstein, Y.Z.: Integrating the Probabilistic Models BM25/BM25F into Lucene. CoRR, abs/0911.5046 (2009)
4. Robertson, S.E., Walker, S.: Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In: Croft, W.B., van Rijsbergen, C.J. (eds.) SIGIR, pp. 232–241. ACM/Springer (1994)
5. Sakai, T., Kando, N., Lin, C.-J., Mitamura, T., Shima, H., Ji, D., Chen, K.-H., Nyberg, E.: Overview of the NTCIR-7 ACLIA IR4QA Task (2008)
6. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. ACM Commun. 18(11), 613–620 (1975)

A Trainable Multi-factored QA System

Radu Ion, Dan Ştefănescu, Alexandru Ceauşu, Dan Tufiş,
Elena Irimia, and Verginica Barbu Mititelu

Research Institute for Artificial Intelligence, Romanian Academy
13, Calea 13 Septembrie, Bucharest 050711, Romania
{radu,danstef,aceausu,tufis,elena,vergi}@racai.ro

Abstract. This paper reports on the construction and testing of a new Question Answering (QA) system, implemented as an workflow which builds on several web services developed at the Research Institute for Artificial Intelligence (RACAI).The evaluation of the system has been independently done by the organizers of the Romanian-Romanian task of the ResPubliQA 2009 exercise and has been rated the best performing system with the highest improvement due to the NLP technology over a baseline state-of-the-art IR system. We describe a principled way of combining different relevance measures for obtaining a general relevance (to the user's question) score that will serve as the sort key for the returned paragraphs. The system was trained on a specific corpus, but its functionality is independent on the linguistic register of the training data. The trained QA system that participated in the ResPubliQA shared task is available as a web application at http://www2.racai.ro/sir-resdec/.

1 Introduction

Looking back the at the QA track of the Text Analysis Conference in 2008 (http://www.nist.gov/tac/) we see that although the QA systems are still very complicated (being basically complex architectures of IR and NLP modules), there are efforts of reducing this complexity in the favor of developing *trainable and principled* QA systems. In this respect, the statistical component of both document scoring and answer extraction is back in business as in IBM's statistical QA systems in TREC-9 [4] and onwards. For instance, Heie et al. [1] use a language model based approach to sentence retrieval for QA in which every sentence is scored according to the probability P(Q|S) of generating *a specific query* Q (a set of terms) for the respective answer sentence S (also a set of terms).

Following the principle of trainability and extensibility, we have devised a QA system that combines a set of snippet relevance scores in a principled way. We were inspired by the Minimum Error Rate Training (MERT) optimization from [5] where a set of weights are trained for a set of features that are supposed to characterize the translation task. In our case, we considered training a set of weights for a set of snippet features that express the relevance of that snippet to the user's question. It also is the case that using a linear combination of different relevance measures (probabilities or simply scores) to provide a unique relevance measure of the sentence or paragraph

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 257–264, 2010.

is the de facto choice in recent QA systems [1, eq. 3; 9, eq. 2]. The only impediment in using MERT is that when trying to optimize the response of the QA system on a test set of N questions, for each question having M snippets returned that are to be globally scored with m parameters with a 10^{-p} precision, there are exactly $M \cdot N \cdot \binom{m + 10^p - 1}{10^p - 1}$ summations of the type equation 1 below shows. In this case, in order to determine the value of the parameters and keeping the time complexity in reasonable limits, one should implement a hill climbing algorithm, setting initial values for the parameters with $p = 1$ and then increase the value of p until the peak of the hill is reached.

In what follows, we will present our ResPubliQA QA system that has been developed according to the principles described above. We will describe the training procedure and the QA algorithm and we will evaluate its performances within the ResPubliQA environment.

2 The QA System

The corpus to be indexed is a subset of the JRC-Acquis comprising of 10714 documents conforming to the TEI format specifications[1]. We only took the body of the document into consideration when extracting the text to be indexed. This text has been preprocessed by TTL and LexPar [8] to obtain POS tagging, lemmatization, chunking and dependency linking.

The body part of one JRC-Acquis document is divided into paragraphs, the unit of text required by the ResPubliQA task to be returned as the answer to the user's question. The specifications of this task define five possible types of questions: "factoid", "definition", "procedure", "reason" and "purpose". The classes "reason" and "purpose" were merged into a port-manteau class "reason-purpose" because we found that our classifier made an unreliable distinction between the two initial classes. By labeling the paragraphs with the type of the expected answer we reduced the complexity of the IR problem: given a query, if its type is correctly identified, the answer is searched through only a portion of the full corpus. We used maximum entropy for paragraph classification. For feature selection we differentiated between clue words, morpho-syntactical, punctuation, orthographical and sentence length related features. The classifier was trained on 800 manually labeled paragraphs from the JRC-Acquis and performs with approximately 94%.

The JRC-Acquis documents are manually classified using the EUROVOC thesaurus[2] that has more than 6000 terms hierarchically organized. Considering the fact that the technical terms occurring in the JRC-Acquis were supposed to be translated using the EUROVOC terms, the term list of our tokenizer was extended so that these terms would be later recognized. If a term is identified, it counts as a single lexical token as in "*adunare parlamentară*" ("parliamentary assembly").

[1] http://www.tei-c.org/Guidelines/
[2] http://europa.eu/eurovoc/

The RACAI's QA system is practically a workflow built on top of our NLP web services. It's a trainable system that uses a linear combination of relevance features scores $\boldsymbol{s_i}$ to obtain a global relevance (to the question) measure $\boldsymbol{S(p)}$ which will be used as the sort key:

$$S(p) = \sum \lambda_i s_i \ , \quad \sum \lambda_i = 1 \tag{1}$$

where $\boldsymbol{s_i}$ is one of the following feature scores ($\boldsymbol{s_i} \in [\mathbf{0}, \mathbf{1}]$):

1. an indicator function that is 1 if the guessed class of the question is identical to that of the candidate paragraph or 0 otherwise (let's call this score s_1);
2. a lexical chains based score computed between lemmas of the candidate paragraph and lemmas of the question (s_2);
3. a BLEU-like [6] score that will give more weight to paragraphs that contain keywords from the question *much* in the same order as they appear in the question (s_3);
4. the paragraph and document scores as returned by the search engine[3] (s_4 and s_5).

When the QA system receives an input question, it first calls the TTL web service[4] to obtain POS tagging, lemmatization and chunking. Then, it calls the question classifier[5] to learn the question class after which two types of queries are computed[6]. Both queries may contain the question class as a search term to be matched with the class of candidate paragraphs. The search engine[7] will return two lists L_1 and L_2 of at most 50 paragraphs that will be sorted according to the eq. 1. The answer is a paragraph p from both L_1 and L_2 for which

$$\underset{p}{\operatorname{argmin}} \left[\operatorname{rank}_1(p) + \operatorname{rank}_2(p)\right], \qquad \operatorname{rank}_{1,2}(p) \leq K, \qquad K \leq 50 \tag{2}$$

where $\operatorname{rank}_j(p)$ is the rank of paragraph p in L_j. Experimenting with different values for K on an in-house developed 200 questions test set (see below), we determined that the best value for K is 3. When such a common paragraph does not exist, the system returns the no answer (NOA) string.

Our QA system is trainable in the sense that the weights (λ_i) that we use to combine our relevance features scores are obtained through a MERT-like optimization technique.

Since the development question set comprised of only 20 questions, we proceeded to the enlargement of this test set (having the 20 questions as examples). We produced another 180 questions to obtain a new development set of 200 questions simply by randomly selecting documents from the JRC-Acquis corpus and reading them. For each question we provided the ID of the paragraph that contained the right answer and the question class. The training procedure consisted of:

---

[3] We used the Lucene search engine (`http://lucene.apache.org`).

[4] `http://ws.racai.ro/ttlws.wsdl`

[5] `http://shadow.racai.ro/JRCACQCWebService/Service.asmx?WSDL`

[6] One of the query computation algorithms is also implemented as a web service and it is available at http://shadow.racai.ro/QADWebService/Service.asmx?WSDL

[7] `http://www.racai.ro/webservices/search.asmx?WSDL`

1. running the QA system on these 200 questions and retaining the first 50 paragraphs for each question according to the paragraph score given by the search engine (s_4);
2. obtaining for each paragraph the set of 5 relevance scores, $s_1 \ldots s_5$;
3. for each combination of λ parameters with $\sum_{i=1}^{5} \lambda_i = 1$ and increment step of 10^{-2}, compute the Mean Reciprocal Rank (MRR) of the 200 question test set by sorting the list of returned paragraphs for each question according to eq. 1;
4. retaining the set of λ parameters for which we obtain the maximum MRR value.

The two QA systems (each one corresponding to specific algorithm of query generation) were individually optimized with no regard to NOA strings and we added the combination function (eq. 2) in order to estimate the confidence in the chosen answer (an optional requirement of the ResPubliQA task).

The first algorithm of query generation (the **TFIDF query algorithm**) considers all the content words of the question (nouns, verbs, adjectives and adverbs) out of which it constructs a disjunction of terms (which are lemmas of the content words) with the condition that the TFIDF of the given term t is above a certain threshold:

$$\mathrm{TFIDF}(t) = (1 + \ln(f_{\mathrm{t}})) \cdot \ln\left(\frac{D}{f_{\mathrm{d}}}\right) \tag{3}$$

in which 'ln' is the natural logarithm, f_{t} is the term frequency in the entire corpus, f_{d} is the number of documents in which the term appears and D is the number of documents in our corpus, namely 10714 (if f_{t} is 0, f_{d} is also 0 and the whole measure is 0 by definition). The rationale behind this decision is that there are certain terms that are very frequent and also very uninformative.

The second algorithm of query generation (the **chunk-based query algorithm**) also uses the TTL preprocessing of the question. As in the previous version [2], the algorithm takes into account the noun phrase (NP) chunks and the main verbs of the question. For each NP chunk, two (instead of one) query terms are constructed: (i) one term is a query expression obtained by concatenating the lemmas of the words in the chunk and having a boost equal to the number of those words, and (ii) the other one is a Boolean query in which all the different lemmas of the words in the chunk are joined by the conjunction operator. For example an "*a b c*" chunk generates the following two queries: `"l(a) l(b) l(c)"^3` and `(l(a) AND l(b) AND l(c))` where `l(w)` is the lemma for the `w` word. For each chunk of length n, we generate all the sub-chunks of length $n-1, n \geq 2$ (i.e. "*a b*" and "*b c*") and apply the same steps.

As already stated, the QA system uses a linear combination of relevance features scores (eq. 1) to score a given candidate paragraph as to the possibility of containing the answer to the question. The BLUE-like similarity measure (s_3) between the question and one candidate paragraph stems from the fact that there are questions that are formulated using a high percentage of words in the order that they appear in the answer containing paragraph. BLEU [6] is a measure that counts *n*-grams from one candidate translation in one or more reference translations. We use the same principle and count *n*-grams from the question in the candidate paragraph but here is where the

similarity to BLEU ends. Our *n*-gram processing counts only content word *n*-grams (content words are not necessarily adjacent). Actually, an *n*-gram is a sliding window of question content word lemmas of a maximum length equal to the length of the question (measured in content words) and a minimum length of 2.

Due to the lack of space, we cannot go into further details regarding the lexical chains paragraph relevance score or question classification but for a detailed description of the entire QA system, we refer the reader to [3].

3 Evaluations

Each query produces a different set of paragraphs when posed to the search engine thus allowing us to speak of two different QA systems. We applied the training procedure described in the previous section on our 200 questions test set with each system and ended up with the following values for the λ parameters:

Table 1. Parameters for paragraph score weighting

	λ_1	λ_2	λ_3	λ_4	λ_5
The TFIDF query algorithm	0.22	0.1	0.1	0.19	0.39
The chunk query algorithm	0.32	0.14	0.17	0.25	0.12

With these parameters, each system was presented with the official ResPubliQA 500 questions test set. For each question, each system returned 50 paragraphs that were sorted according to eq. 1 using parameters from Table 1. Table 2 contains the official evaluations [9] of our two runs, ICIA091RORO and ICIA092RORO. The first run, officially rated with the fourth c@1 score, corresponds to running the two QA systems with queries exactly as described. The second run, officially rated with the best c@1 score, was based on queries that contained the class of the question. When we constructed the index of paragraphs we added a field that kept the paragraph class. This tweak brought about a significant improvement in both accuracy and c@1 measure as Table 2 shows.

A basic assumption made by organizers when using the c@1 evaluation score apparently was that the accuracy of detecting questions with wrong answers, for which the QA systems should refrain from providing a misleading reply but output a NOA, was more or less the same with the accuracy of effectively answered questions. We checked this assumption for our systems and found that this was a reasonable hypothesis, although, in our case the precision of detection the answers with a likely wrong answer (ICIA091RORO=70%, ICIA092RORO=86%) was much higher than the precision in providing correct answers: ICIA091RORO = 47%, ICIA092RORO = 52%). Therefore the c@1 measure is a rather conservative/pessimistic measure. If we had decided to answer all the questions the accuracy would have been better (7% for the first run and 4% for second run) but the c@1 score would have decreased.

A very interesting evaluation performed by the organizers was to estimate the accuracy improvement of the NLP QA systems as compared to language-specific baseline IR systems ([7]). According to this new evaluation both ICIA runs received the highest scores out of 20 evaluated runs.

In order to estimate how many correct answers would have been returned instead of the NOA strings, we ran our QA systems on the 500 questions test set with the exact same settings as per Table 1 obtaining runs ICIA091RORO-NO-NOA and ICIA092RORO-NO-NOA that were combinations resulting from setting K to the maximum allowed value of 50 in eq. 2 (this way, eq. 2 always returned a paragraph thus eliminating the presence of the NOA string). Then, for the two pairs of runs (ICIA091RORO, ICIA091RORO-NO-NOA) and (ICIA092RORO, ICIA092RORO-NO-NOA) we checked the status of every NOA string by seeing if the corresponding answer was correct or not (using the official Gold Standard of Answers of the Romanian-Romanian ResPubliQA task that we got from the organizers). These results are also displayed by Table 2.

Table 2. RACAI official results and the reevaluated results when NOA strings have been replaced by actual answers

	ICIA091RORO	**ICIA092RORO**
ANSWERED	393	344
UNANSWERED	107	156
ANSWERED with RIGHT candidate	237	260
ANSWERED with WRONG candidate	156	84
c@1 measure	0.58	0.68
Overall accuracy	0.47	0.52
UNANSWERED with RIGHT candidate	**32**	**21**
UNANSWERED with WRONG candidate	**75**	**135**
UNANSWERED with EMPTY candidate	**0**	**0**
Predicted wrong ans./Actual wrong ans. in the initial NOA questions	**50/75**	**81/135**
Reevaluated overall accuracy	**0.54**	**0.56**

Table 2 lists the results of our QA system that used a single set of λ parameters for every question class. We hypothesized that training different sets of λ parameters for each QA system and for each question class would yield improved results. We experimented with our 200 questions test set and trained different sets of parameters (with the increment step of 0.05 to reduce the time complexity) for each question class. Table 3 presents the trained values for the parameters.

Running our QA systems over the 500 questions test set, sorting the returned paragraphs for each question using the set of parameters trained on the question's class and combining the results with K = 50 to remove the NOA strings, we obtained an overall accuracy of **0.5774**. This confirmed our hypothesis that class-trained parameters would improve the performance of the system.

Table 3. Different parameters trained for different classes

		λ_1	λ_2	λ_3	λ_4	λ_5
The TFIDF query algorithm	Factoid	0.1	0	0.2	**0.4**	0.3
	Definition	**0.2**	0.15	0.05	0.15	0.45
	Reason	0.1	0	0.15	0.3	0.45
	Procedure	0.1	0	0.15	0.15	**0.6**
The chunk query algorithm	Factoid	0.15	0	0.3	0.3	0.25
	Definition	0.05	**0.5**	0.15	0.1	0.2
	Reason	**0.2**	0	**0.4**	0.2	0.2
	Procedure	0.15	0.1	0.25	0.2	0.3

4 Conclusions

The CLEF campaign has gone long way into the realm of Information Retrieval and Extraction. Each year, the evaluation exercise showed its participants how to test and then, how to improve their systems. The competitive framework has motivated systems designers to adopt daring solutions and to experiment in order to obtain the best result. However, we should not forget that we are building these IR and QA systems primarily to help people to search for the information they need. In this regard, it would be very helpful that future shared tracks would require that the QA systems be available on the Internet. Another dimension in which we can extend the usefulness of the evaluation is the scoring formulas. The requirement that only the first answer is to be evaluated is a little harsh given the willingness of the average user to inspect, say at least 5 top returned results.

We want to extend the present QA system so that it can cross-lingually answer question from English or Romanian in either English or Romanian. We have already processed the English side of the JRC-Acquis and, given that we have several functional Example-Based and Statistical Machine Translation Systems, we may begin by automatically translating either the natural language question or the generated query. Then the combination method expressed by eq. 2 would probably yield better results if applied on English and Romanian paragraph lists since a common paragraph means the same information found via two different languages. This estimation is strengthen by the analysis made by the ResPubliQA organizers, according to which 99% of questions have been correctly answered by at least one system in at least one language.

The principal advantage of this approach to QA is that one has an easily extensible and trainable QA system. If there is another way to assess the relevance of a paragraph to a given question, simply add another parameter that will account for the importance of that measure and retrain. We believe that in the interest of usability, understandability, adaptability to other languages and, ultimately progress, such principled methods are to be preferred over the probably more accurate but otherwise almost impossible to reproduce methods.

Acknowledgements. The work reported here is funded by the SIR-RESDEC project, financed by the Ministry of Education, Research and Innovation under the grant no 11-007.

References

1. Heie, M.H., Whittaker, E.W.D., Novak, J.R., Mrozinski, J., Furu, S.: TAC2008 Question Answering Experiments at Tokyo Institute of Technology. In: Proceedings of the Text Analysis Conference (TAC 2008), November 17-19. National Institute of Standards and Technology, Gaithersburg, (2008)
2. Ion, R., Ştefănescu, D., Ceauşu, A., Tufiş, D.: RACAI's QA System at the Romanian-Romanian Multiple Language Question Answering (QA@CLEF2008) Main Task. In: Nardi, A., Peters, C. (eds.) CLEF 2008. LNCS, vol. 5706, p. 10. Springer, Heidelberg (2009)
3. Ion, R., Ştefănescu, D., Ceauşu, A., Tufiş, D., Irimia, E., Barbu Mititelu, V.: A Trainable Multi-factored QA System. In: Peters, C., et al. (eds.) Working Notes for the CLEF 2009 Workshop, Corfu, Greece, p. 14 (October 2009)
4. Ittycheriah, A., Franz, M., Zhu, W.J., Ratnaparkhi, A., Mammone, R.J.: IBM's Statistical Question Answering System. In: Proceedings of the 9th Text Retrieval Conference (TREC-9), Gaithersburg, Maryland, November 13-16, pp. 229–235 (2000)
5. Och, F.J.: Minimum Error Rate Training in statistical machine translation. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Sapporo, Japan, July 07-12, pp. 160–167 (2003)
6. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the ACL 2002, 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, pp. 311–318 (July 2002)
7. Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, Á., Forăscu, C., Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P.: Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. In: Peters, C., et al. (eds.) Working Notes for the CLEF 2009 Workshop, Corfu, Greece, October 2009, p. 14 (October 2009)
8. Tufiş, D., Ion, R., Ceauşu, A., Ştefănescu, D.: RACAI's Linguistic Web Services. In: Proceedings of the 6th Language Resources and Evaluation Conference - LREC 2008. ELRA - European Language Ressources Association, Marrakech, Morocco (May 2008)
9. Wiegand, M., Momtazi, S., Kazalski, S., Xu, F., Chrupała, G., Klakow, D.: The Alyssa System at TAC QA 2008. In: Proceedings of the Text Analysis Conference (TAC 2008), November 17-19. National Institute of Standards and Technology, Gaithersburg (2008)

Extending a Logic-Based Question Answering System for Administrative Texts

Ingo Glöckner[1] and Björn Pelzer[2]

[1] Intelligent Information and Communication Systems Group (IICS),
University of Hagen, 59084 Hagen, Germany
ingo.gloeckner@fernuni-hagen.de
[2] Department of Computer Science, Artificial Intelligence Research Group
University of Koblenz-Landau, Universitätsstr. 1, 56070 Koblenz
bpelzer@uni-koblenz.de

Abstract. LogAnswer is a question answering (QA) system for German that uses machine learning for integrating logic-based and shallow (lexical) validation features. For ResPubliQA 2009, LogAnswer was adjusted to specifics of administrative texts, as found in the JRC Acquis corpus. Moreover, support for a broader class of questions relevant to the domain was added, including questions that ask for a purpose, reason, or procedure. Results confirm the success of these measures to prepare LogAnswer for ResPubliQA, and of the general consolidation of the system. According to the *C@1/Best IR baseline* metric that tries to abstract from the language factor, LogAnswer was the third best of eleven systems participating in ResPubliQA. The system was especially successful at detecting wrong answers, with 73% correct rejections.

1 Introduction

LogAnswer is a question answering system that uses logical reasoning for validating possible answer passages and for extracting answer phrases.[1] It was first evaluated in QA@CLEF 2008 [1]. The ResPubliQA task [2] posed some new challenges for LogAnswer:

- The JRC Acquis[2] corpus contains administrative texts that are hard to parse. Since logical processing in LogAnswer requires syntactic-semantic analyses, the parser had to be adjusted to JRC Acquis, and a graceful degradation of results had to be ensured when parsing fails.
- LogAnswer had to be extended to detect the new PURPOSE, PROCEDURE and REASON questions of ResPubliQA and to find matching answers.
- While LogAnswer used to work strictly sentence-oriented, ResPubliQA now expects a whole paragraph to be found that best answers a question.

[1] Funding by the DFG (Deutsche Forschungsgemeinschaft), FU 263/12-1, HE 2847/10-1 (LogAnswer) is gratefully acknowledged. Thanks to Tim vor der Brück for his n-gram recognizer, and to Sven Hartrumpf for adapting the WOCADI parser.

[2] See http://langtech.jrc.it/JRC-Acquis.html

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 265–272, 2010.

– The main evaluation metric of ResPubliQA, the c@1 score, rewards QA systems that prefer not answering over giving wrong answers. LogAnswer computes a quality score that includes a logical validation. A threshold for cutting off poor answers had to be determined that optimizes the c@1 metric.

The overall goal of our participation in QA@CLEF was that of evaluating the various improvements of the system; this includes the refinements of LogAnswer based on lessons from QA@CLEF 2008 and the extensions for ResPubliQA.

In the paper, we first introduce the LogAnswer system. Since many aspects of LogAnswer are already described elsewhere [1,4,5], we focus on novel developments added for ResPubliQA. We then detail the results of LogAnswer and show the effectiveness of the measures taken to prepare LogAnswer for ResPubliQA.

2 Preparing LogAnswer for the ResPubliQA Task

2.1 Overview of the LogAnswer System

LogAnwer uses the WOCADI parser [6] for a deep linguistic analysis of texts and questions. After retrieving 100 candidate snippets, the system tries to prove the logical question representation from that of each candidate passage to be validated and from its background knowledge [5]. For better robustness to knowledge gaps, the prover is embedded in a relaxation loop that skips non-provable literals until a proof of the reduced query succeeds. Adding 'shallow' criteria (e.g. lexical overlap) also improves robustness. A machine learning (ML) approach computes a quality score for the text passage from this data. If several passages support an answer, then their quality scores are combined into the final answer score [7].

2.2 Improvements of Document Analysis and Indexing

Optimization of the WOCADI Parser for Administrative Language. WOCADI finds a full parse for more than half of the sentences in the German Wikipedia, but this number was only 26.2% for JRC Acquis (partial parse rate: 54%). In order to find more parsing results, a technique for reconstructing German characters ä, ö, ü and ß from ae, oe etc. was added, and the case sensitivity of the parser was switched off for sentences with many fully capitalized words. Another problem are the complex references to (sections of) regulations etc. in administrative texts, e.g. *"(EWG) Nr. 1408/71 [3]"*. We trained an n-gram recognizer using the current token and up to three previous tokens. The probability that a token belongs to a complex document reference is estimated by a log-linear model based on the probabilities for unigrams, bigrams, trigrams and four-grams. The representation of the complex name is then filled into the parsing result.

The changes to WOCADI achieved a relative gain of 12.6% in the rate of full parses, and of 5.6% for partial parses. Still, only 29.5% of the sentences in JRC Acquis are assigned a full parse. So, the extension of LogAnswer by techniques for handling non-parseable sentences had to be enforced for ResPubliQA.

Interestingly, the questions in the ResPubliQA test set were much easier to parse than the texts in the corpus: For the ResPubliQA questions, WOCADI had a full parse rate of 90% and partial parse rate (with chunk parses) of 96.4%.

Indexing Sentences with a Failed Parse. In the first LogAnswer prototype, only sentences with a full parse were indexed. But JRC Acquis is hard to parse, so we have now included sentences with a failed or incomplete parse as well. Since LogAnswer also indexes the possible answer types found in the sentences [1], the existing solution for extracting answer types had to be extended to recognize expressions of these types in arbitrary sentences.

We also complemented the special treatment of regulation names described above by a method that helps for non-parseable sentences. The tokenization of the WOCADI parser was enriched by two other tokenizers: the GermanAnalyzer of Lucene, and a special tokenizer for recognizing email addresses and URLs. New tokens found by these special tokenizers were also added to the index.

Support for New Question Categories. Trigger words (and more complex patterns applied to the morpho-lexical analysis of sentences) were added for recognizing sentences that describe methods, procedures, reasons, purposes, or goals. By indexing these types, LogAnswer can focus retrieval to matching sentences.

Beyond Indexing Individual Sentences. One novel aspect of ResPubliQA was the requirement to submit answers in the form of full paragraphs. This suggests the use of paragraph retrieval or of other means for finding answers when the relevant information is scattered over several sentences. In addition to its sentence index, LogAnswer now offers a paragraph-level and a document-level index. Moreover a special treatment for anaphoric pronouns based on the CORUDIS coreference resolver [6] was added. Whenever CORUDIS finds an antecedent for a pronoun, the antecedent is used for enriching the description of the considered sentence in the index. So, if the pronoun 'es' in a sentence refers to 'Spanien' (Spain), then 'Spanien' is also added to the index.

2.3 Improvements of Question Processing

Syntactic-Semantic Parsing of the Question. The linguistic analysis of questions also profits from the adjustments of WOCADI to administrative texts. References to (sections of) legal documents in a question are treated in a way consistent with the treatment of these constructions in answer paragraphs. Similarly, the additional tokenizers used for segmenting the texts are also applied to the question in order to generate a matching retrieval query.

Refinement of Question Classification. LogAnswer uses a rule-based question classification for recognizing question categories. Its system of 165 classification rules now also covers the new categories PROCEDURE, REASON, PURPOSE of ResPubliQA. The new classification rules have two effects: (a) the question category is identified, so that retrieval can focus on suitable text passages; and (b) expressions like 'What is the reason' or 'Do you know' can be deleted from the descriptive core of the question used for retrieval and reasoning.

Querying the Enriched Index. The retrieval step profits from all improvements described in Sect. 2.2. Since many validation features of LogAnswer are still sentence-based, the sentence-level index was queried for each question in order to fetch the logical representation of 100 candidate sentences. For experiments on the effect of paragraph-level and document-level indexing, the 200 best paragraphs and the 200 best documents for each question were also retrieved.

Features for Candidate Quality. The validation features described in [5] (e.g. lexical overlap, answer type check...) were used for assessing the quality of retrieved passages. The definition of these features was extended to sentences with a failed parse in order to improve validation results in this case.

Estimation of Validation Scores. One of our lessons from QA@CLEF08 was the inadequacy of the first ML approach of LogAnswer for computing quality scores for the retrieved passages. We thus developed a new solution using rank-optimizing decision trees [5]. The result was a 50% accuracy gain of LogAnswer on the QA@CLEF 2008 test set [1]. The new models were also used for ResPubliQA. The obtained scores for individual sentences are then aggregated [7].

Optimization of the c@1 Score. The threshold θ for accepting the best answer (or refusing to answer) was chosen such as to optimize the c@1 score of LogAnswer on the ResPubliQA development set: The questions were translated into German, and LogAnswer was run on the translations. $\theta = 0.08$ was then identified as the optimum threshold, achieving a c@1 score of 0.58 on the training set. Once a retrieved sentence with top rank is evaluated better than θ, the paragraph that contains the sentence becomes the final LogAnswer result for the given question.

3 Results on the ResPubliQA 2009 Test Set for German

The results of LogAnswer in ResPubliQA 2009 and the two official baseline results [2] are shown in Table 1. The *loga091dede* run was obtained from the standard configuration of LogAnswer with full logic-based processing, while *loga092dede* was generated with the prover switched off. It simulates the case that all retrieved passages have a failed parse. Considering the number of questions with a correct paragraph at top rank, the logic-based run *loga091dede* performed best, closely followed by the shallow LogAnswer run and then the baseline runs *base092dede* and *base091dede*. LogAnswer clearly outperforms both baselines with respect to the c@1 score that includes validation quality. It was also good at detecting wrong answers: The decision to reject answers with a low validation was correct in 73% of the cases.

3.1 Strengths and Weaknesses of LogAnswer

A breakdown of results by question category is shown in Table 2. LogAnswer was best for FACTOID and REASON questions. PROCEDURE and DEFINITION results were poor in both LogAnswer runs.

Table 1. Results of LogAnswer in ResPubliQA. #right cand. is the number of correct paragraphs at top rank before applying θ, and accuracy = #right cand./#questions

run	description	#right cand.	accuracy	c@1 score
loga091dede	complete system	202	0.40	0.44
loga092dede	system w/o prover	199	0.40	0.44
base091dede	official baseline 1	174	0.35	0.35
base092dede	official baseline 2	189	0.38	0.38

Table 2. Accuracy by question category

Run	DEFINITION	FACTOID	PROCEDURE	PURPOSE	REASON
	(95)	(139)	(79)	(94)	(93)
loga091dede	0.168	0.547	0.291	0.362	0.570
loga092dede	0.137	0.554	0.291	0.362	0.559

As to definition questions, the training set for learning the validation model (using QA@CLEF 2007/2008 questions) included too few examples for successful application of our ML technique. Thus the model for factoids (also used for REASON etc.) is much better than the model for definition questions.

Another factor is the treatment of references to definitions, e.g.

> *"Permanent pasture" shall mean "permanent pasture" within the meaning of Article 2 point (2) of Commission Regulation (EC) No 795/2004.*

It was not clear to us that such definitions by reference would not be accepted. But the main problem was the form of many definitions in JRC Acquis, e.g.

> *Hop powder: the product obtained by milling the hops, containing all the natural elements thereof.*

Since the retrieval queries for definition questions were built such that only sentences known to contain a definition were returned, many definitions of interest were skipped only because this domain-specific way of expressing definitions was not known to LogAnswer. The solution is making the requirement that retrieved sentences contain a recognized definition an optional part of the retrieval query.

The PROCEDURE result reflects the difficulty of recognizing sentences describing procedures, as needed for guiding retrieval to relevant sentences.

A breakdown of FACTOID results is shown in Table 3. Questions for countries were classified LOCATION or ORG(ANIZATION), depending on the question. OTHER and OBJECT questions were lumped together since LogAnswer does not discern these types. In both LogAnswer runs, LOCATION, ORGANIZATION and PERSON questions clearly worked well.

Table 3. Accuracy by expected answer types for the FACTOID category

Run	COUNT (3)	LOCATION (8)	MEASURE (16)	ORG (14)	OTHER (80)	PERSON (3)	TIME (16)
loga091dede	0.33	0.75	0.56	0.71	0.51	1.00	0.44
loga092dede	0.33	1.00	0.56	0.71	0.50	1.00	0.44

Table 4. Success rate of question classification (class-all is the classification rate for arbitrary questions and class-fp the classification rate for questions with a full parse)

Category	#questions	class-all	#full parse	class-fp
DEFINITION	95	85.3%	93	87.1%
REASON	93	73.3%	82	85.4%
FACTOID	139	70.5%	117	76.9%
PURPOSE	94	67.0%	86	72.1%
PROCEDURE	79	20.3%	72	22.2%
(total)	500	65.6%	450	70.9%

3.2 Effectiveness of Individual Improvements

Recognition of References to Legal Documents. The ResPubliQA test set contains 15 questions with names of legal documents that our n-gram recognizer should find; the recognition rate was 87%. The benefit of a found reference is that the parser has a better chance of analyzing the question, in which case the interpretation of the reference is filled into the semantic representation. Since the recognized entities are indexed, retrieval also profits in this case.

Use of Additional Tokenizers. The special treatment of references to legal documents is only effective for parseable sentences. However, some of these references are also covered by the extra tokenizers that were added to LogAnswer. On the ResPubliQA 2009 test set, this happened for 21 questions. The benefit of analyzing regulation names like *821/68* as one token is, again, the increased precision of retrieval compared to using a conjunction of two descriptors *821* and *68*.

Effectiveness of Changes to the Retrieval Module. For ResPubliQA, sentences with a failed or incomplete parse were added to the index. Considering the 202 correct answer paragraphs in the *loga091dede* run, only 49 of these answers were based on the retrieval of a sentence of the paragraph with a full parse, while 106 stem from a sentence with a chunk parse, and 47 from a sentence with a failed parse (similar for *loga092dede*). Thus, extending the index beyond sentences with a full parse was essential for the success of LogAnswer in ResPubliQA.

Success Rate of Question Classification. Results on the success rate of question classification in LogAnswer are shown in Table 4. The recognition rules apply

to the parse of a question, so the results for questions with a full parse are most informative. The classification was best for DEFINITION, REASON and FACTOID questions. The low recognition rates for PURPOSE and PROCEDURE are due to missing trigger words that signal questions of these types: *'Zielvorstellung'* (objective) was not a known PURPOSE trigger, and *'Verfahren'* (process) was not listed for PROCEDURE. Compounds of triggers were also not recognized, e.g. *Arbeitsverfahren* (working procedure), or *Hauptaufgabe* (main task). The problem can be fixed by adding these trigger words, and by treating compounds of trigger words also as triggers for a question type.

Effect of Question Classification. Since ResPubliQA does not require exact answer phrases, we wondered if recognizing the question category and expected answer type is still needed. We checked *loga091dede* and found that for all question categories except DEFINITION, results were more accurate for questions classified correctly. For definition questions, however, the accuracy for the 14 misclassified questions was 0.36, while for the 81 correctly recognized questions, the accuracy was only 0.14. The two cases differ in the retrieval query: if a definition question is identified, then an obligatory condition is added to the query that cuts off all sentences not known to contain a definition. If a definition question is not recognized, this requirement is missing. This suggests that the obligatory condition should be made an optional part of the retrieval query.

Selection of Acceptance Threshold. In the ResPubliQA runs, a threshold of $\theta = 0.08$ was used for cutting off wrong answers. The optimum for *loga091dede* would have been $\theta = 0.11$ (c@1 score 0.45 instead of 0.44). For *loga092dede*, the optimum would have been $\theta = 0.09$, but it hardly changes the c@1 score. Thus, the method for finding θ (by choosing the threshold that maximizes c@1 on the development set) was effective – it resulted in close-to-optimal c@1 scores.

3.3 Experiments with Paragraph-Level and Document Indexing

One of the features used for determining validation scores is the original retrieval score of the Lucene-based retrieval module of LogAnswer. In order to assess the potential benefit of paragraph-level and document-level indexing, we have experimented with the *irScore* feature. Consider a retrieved candidate sentence c. Then the following variants have been tried: $irScore_s(c)$ (original retrieval score on sentence level), $irScore_p(c)$ (retrieval score of the paragraph containing c), $irScore_d(c)$ (retrieval score of the document containing c), and also the following combinations: $irScore_{ps}(c) = \frac{1}{2}(irScore_p(c) + irScore_s(c))$, $irScore_{ds}(c) = \frac{1}{2}(irScore_d(c) + irScore_s(c))$, $irScore_{dp}(c) = \frac{1}{2}(irScore_d(c) + irScore_p(c))$, and finally $irScore_{dps}(c) = \frac{1}{3}(irScore_d(c) + irScore_p(c) + irScore_s(c))$. See Table 5 for results; the $irScore_s$ configuration corresponds to *loga091dede*. The document-only result ($irScore_d$) shows that either sentence-level or paragraph-level information is needed for selecting correct answers.

Table 5. Experimental Results using Paragraph-Level and Document-Level Indexing

run	#right	cand. accuracy
$irScore_{ps}$	205	0.41
$irScore_{s}$	202	0.40
$irScore_{dps}$	198	0.40
$irScore_{p}$	196	0.40

run	#right	cand. accuracy
$irScore_{dp}$	191	0.39
$irScore_{ds}$	190	0.38
$irScore_{d}$	136	0.28

4 Conclusion

We have presented the LogAnswer QA system. Results confirm that the measures taken to prepare LogAnswer for ResPubliQA were effective. Due to the low parse rate for JRC Acquis, the use of logical reasoning in the first LogAnswer run meant only a slight benefit. On the other hand, the results in the shallow-only run show that the proposed robustness enhancements work. In general, the ResPubliQA results suggest that shallow linguistic processing or even plain passage retrieval can often identify the correct answer, but this may simply reflect that many ResPubliQA questions use the exact wording found in the answer paragraph. LogAnswer is online at `www.loganswer.de`, and in this configuration, it shows the five best answers. We found that for *loga091dede*, 60% of the questions are answered by one of the top-five paragraphs shown to the user. This number will further improve once the identified bugs are fixed, and we then expect LogAnswer to become a very useful tool for searching information in administrative texts.

References

1. Glöckner, I., Pelzer, B.: Combining logic and machine learning for answering questions. In: [3], pp. 401–408
2. Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, A., Forascu, C., Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P.: Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 174–196. Springer, Heidelberg (2010)
3. Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.): CLEF 2008. LNCS, vol. 5706. Springer, Heidelberg (2009)
4. Furbach, U., Glöckner, I., Helbig, H., Pelzer, B.: LogAnswer - A Deduction-Based Question Answering System. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 139–146. Springer, Heidelberg (2008)
5. Furbach, U., Glöckner, I., Pelzer, B.: An application of automated reasoning in natural language question answering. AI Communications (2010) (to appear)
6. Hartrumpf, S.: Hybrid Disambiguation in Natural Language Analysis. Der Andere Verlag, Osnabrück (2003)
7. Glöckner, I.: RAVE: A fast logic-based answer validator. In: [3], pp. 468–471

Elhuyar-IXA: Semantic Relatedness and Cross-Lingual Passage Retrieval

Eneko Agirre[1], Olatz Ansa[1], Xabier Arregi[1], Maddalen Lopez de Lacalle[2], Arantxa Otegi[1], Xabier Saralegi[2], and Hugo Zaragoza[3]

[1] IXA NLP Group, University of the Basque Country. Donostia, Basque Country
{e.agirre,olatz.ansa,xabier.arregi,arantza.otegi}@ehu.es
[2] R&D, Elhuyar Foundation. Usurbil, Basque Country
{maddalen,xabiers}@elhuyar.com
[3] Yahoo! Research. Barcelona, Spain
hugoz@yahoo-inc.com

Abstract. This article describes the participation of the joint Elhuyar-IXA group in the ResPubliQA exercise at QA&CLEF. In particular, we participated in the English–English monolingual task and in the Basque–English cross-lingual one. Our focus has been threefold: (1) to check to what extent information retrieval (IR) can achieve good results in passage retrieval without question analysis and answer validation, (2) to check Machine Readable Dictionary (MRD) techniques for the Basque to English retrieval when faced with the lack of parallel corpora for Basque in this domain, and (3) to check the contribution of semantic relatedness based on WordNet to expand the passages to related words. Our results show that IR provides good results in the monolingual task, that our crosslingual system performs lower than the monolingual runs, and that semantic relatedness improves the results in both tasks (by 6 and 2 points, respectively).

1 Introduction

The joint team was formed by two different groups, on the one hand the Elhuyar Foundation, and on the other hand the IXA NLP group. This collaboration allowed us to tackle the English–English monolingual task and the Basque–English cross-lingual one in the ResPubliQA track.

With respect to the Basque-English task, we met the challenge of retrieving English passages for Basque questions. We tackled this problem by translating the lexical units of the questions into English. The main setback is that no parallel corpus was available for this pair of languages, given that there is no Basque version of the JRC-Acquis collection. So we have explored an approach which does not use parallel corpora when translating queries, which could also be interesting for other less resourced languages. In our opinion, bearing in mind the idiosyncrasy of the European Union, it is worthwhile dealing with the search of passages that answer questions formulated in unofficial languages.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 273–280, 2010.

Question answering systems typically rely on a passage retrieval system. Given that passages are shorter than documents, vocabulary mismatch problems are more important than in full document retrieval. Most of the previous work on expansion techniques has focused on pseudo-relevance feedback and other query expansion techniques. In particular, WordNet has been used previously to expand the terms in the query with little success [2, 3, 4]. The main problem is ambiguity, and the limited context available to disambiguate the word in the query effectively. As an alternative, we felt that passages would provide sufficient context to disambiguate and expand the terms in the passage. In fact, we do not do explicit word sense disambiguation, but rather apply a state-of-the-art semantic relatedness method [5] in order to select the best terms to expand the documents.

2 System Overview

2.1 Question Pre-processing

We analysed the Basque questions by re-using the linguistic processors of the *Ihardetsi* question-answering system [1]. This module uses two general linguistic processors: the lemmatizer/tagger named *Morfeus* [6], and the Named Entity Recognition and Classification (NERC) processor called *Eihera* [7]. The use of the lemmatizer/tagger is particularly suited to Basque, as it is an agglutinative language. It returns only one lemma and one part of speech for each lexical unit, which includes single word terms and multiword terms (MWTs) (those included in the Machine Readable Dictionary (MRD) introduced in the next subsection). The NERC processor, *Eihera*, captures entities such as *person*, *organization* and *location*. The numerical and temporal expressions are captured by the lemmatizer/tagger. The questions thus analyzed are passed to the translation module.

English queries were tokenized without further analysis.

2.2 Translation of the Query Terms (Basque-English Runs)

Once the questions had been linguistically processed, we translated them into English. Due to the scarcity of parallel corpora for a small language or even for big languages in certain domains, we have explored a MRD-based method. These approaches have inherent problems, such as the presence of ambiguous translations and also out-of-vocabulary (OOV) words. To tackle these problems, some techniques have been proposed such as structured query-based techniques [8, 9] and concurrences-based techniques [10, 11]. These approaches have been compared for Basque by obtaining best MAP (Mean Average Precision) results with structured queries [12]. However, structured queries were not supported in the retrieval algorithm used (see Section 2.3), so we adopted a concurrences-based translation selection strategy.

The translation process designed comprises two steps and takes the keywords (Name Entities, MWTs and single words tagged as noun, adjective or verb) of the question as source words.

In the first step the translation candidates of each source word are obtained. The translation candidates for the lemmas of the source words are taken from a bilingual eu-en MRD composed from the Basque-English *Morris* dictionary[1], and the *Euskalterm* terminology bank[2] which includes 38,184 MWTs. After that, OOV words and ambiguous translations are dealt with. The number of OOV words quantified out of a total of 421 keywords for the 77 questions of the development set was 42 (10%). Nevertheless, it must be said that many of these OOV words were wrongly tagged lemmas and entities. We deal with OOV words by searching for their cognates in the target collection. The cognate detection is done in two phases. Firstly, we apply several transliteration rules to the source word. Then we calculate the Longest Common Subsequence Ratio (LCSR) among words with a similar length (+-10%) from the target collection (see Figure 1). The ones which reach a previously established threshold (0.9) are selected as translation candidates. The MWTs that are not found in the dictionary are translated word by word, as we realized that most of the MWTs could be translated correctly in that way, exactly 91% of the total MWTs identified by hand in the 77 development questions.

err- ---> r-	*erradioterapeutiko=radioterapeutiko*
k ---> *c*	*radioterapeutiko=radioterapeutico*
LCSR(radioterapeutico, radioterapeutic) = 0.9375	

Fig. 1. Example of cognate detection

In the second step, we select the best translation of each source keyword according to an algorithm based on target collection concurrences. This algorithm sets out to obtain the translation candidate combination that maximizes their global association degree. We take the algorithm proposed by Monz and Dorr [11].

Initially, all the translation candidates are equally likely. Assuming that t is a translation candidate of the set of all candidates $tr(s_i)$ for a query term s_i given by the MRD, then:

Initialization step:

$$w_T^0(t \mid s_i) = \frac{1}{|tr(s_i)|} \tag{1}$$

In the iteration step, each translation candidate is iteratively updated using the weights of the rest of the candidates and the weight of the link connecting them.

Iteration step:

$$w_T^n(t \mid s_i) = w_T^{n-1}(t \mid s_i) + \sum_{t' \in inlink(t)} w_L(t,t') \cdot w_T^{n-1}(t' \mid s_i) \tag{2}$$

[1] English/Basque dictionary including 67,000 entries and 120,000 senses.

[2] Terminological dictionary including 100,000 terms in Basque with equivalences in Spanish, French, English and Latin.

where $inlink(t)$ is the set of translation candidates that are linked to *t*, and $w_L(t,t')$ is the association degree between *t* and *t'* on the target passages measured by Log-likelihood ratio. These concurrences were calculated by taking the target passages as window.

After re-computing each term weight they are normalized.

Normalization step:

$$w_T^n(t \mid s_i) = \frac{w_T^n(t \mid s_i)}{\sum_{m=1}^{|tr(s_i)|} w_T^n(t_{i,m} \mid s_i)} \tag{3}$$

The iteration stops when the variations of the term weights become smaller than a predefined threshold.

We have modified the iteration step by adding a factor $w_F(t,t')$ to increase the association degree $w_I(t,t')$ between translation candidates *t* and *t'* whose corresponding source words $so(t), so(t')$ are close to each other (distance *dis* in words is low) in the source query *Q*, or even belong to the same Multi-Word Unit ($smw(so(t), so(t'))=1$). As the global association degree between translation candidates is estimated from the association degree of pairs of candidates, we score positively these two characteristics when the association degree for a pair of candidates is calculated. Thus, the modified association degree $w'_L(t,t')$ between *t* and *t'* will be calculated in this way:

$$w'_L(t,t') = w_L(t,t') w_F(t,t') \tag{4}$$

$$w_F(t,t') = \frac{\max_{si,sj \in Q} dis(s_i, s_j)}{dis(so(t), so(t'))} \cdot 2^{smw(so(t),so(t'))} \tag{5}$$

$$smw(s,s') = \begin{cases} 1 & \{s,s'\} \subseteq Z \quad \text{where } Z \in MWU \\ 0 & \end{cases} \tag{6}$$

2.3 Passage Retrieval

The purpose of the passage retrieval module is to retrieve passages from the document collection which are likely to contain an answer. The main feature of this module is that the passages are expanded based on their related concepts, as explained in the following sections.

2.3.1 Document Preprocessing and Application of Semantic Relatedness

Given that the system needs to return paragraphs, we first split the document collection into paragraphs. Then we lemmatized and part-of-speech (POS) tagged those passages using the OpenNLP open source software[3].

[3] http://opennlp.sourceforge.net/

After preprocessing the documents, we expanded the passages based on semantic relatedness. To this end, we used UKB[4], a collection of programs for performing graph-based Word Sense Disambiguation and lexical similarity/relatedness using a pre-existing knowledge base [5], in this case WordNet 3.0.

Given a passage (represented using the lemmas of all nouns, verbs, adjectives and adverbs), UKB returns a vector of scores for concepts in WordNet. Each of these concepts has a score, and the higher the score, the more related the concept is to the given passage. Given the list of related concepts, we took the highest-scoring 100 concepts and expanded them to all variants (words that lexicalize the concepts) in WordNet. An example of a document expansion is shown in Figure 2.

We applied the expansion strategy only to passages which had more than 10 words (half of the passages), for two reasons: the first one is that most of these passages were found not to contain relevant information for the task (e.g. "Article 2", "Having regard to the proposal from the Commission" or "HAS ADOPTED THIS REGULATION"), and the second is that we thus saved some computation time.

2.3.2 Indexing

We indexed the new expanded documents using the MG4J search-engine [13]. MG4J makes it possible to combine several indices over the same document collection. We created one index for the original words and another one with the variants for the most related 100 concepts. This way, we were able to use the original words only, or alternatively, to also include the expanded words during the retrieval. Porter stemmer was used.

2.3.3 Retrieval

We used the BM25 ranking function with the following parameters: 1.0 for *k1* and 0.6 for *b*. We did not tune these parameters. MG4J allows multi-index queries, where one can specify which of the indices one wants to search in, and assign different weights to each index. We conducted different experiments, by using only the index made of original words and also by using the index with the expansion of concepts, giving different weights to the original words and the expanded concepts. The weight of the index which was created using the original words from the passages was 1.00 for all the runs. 1.00 was also the weight of the index that included the expanded words for the monolingual run, but it was 1.78 for the bilingual run. These weights were fixed following a training phase with the English development questions provided by the organization, and after the Basque questions had been translated by hand (as no development Basque data was released).The submitted runs are described in the next section.

3 Description of Runs

We participated in the English-English monolingual task and the Basque-English cross-lingual task. We did not analyze the English queries for the monolingual run,

[4] The algorithm is publicly available at `http://ixa2.si.ehu.es/ukb/`

and we just removed the stopwords. For the bilingual runs, we first analyzed the questions (see Section 2.1), then we translated the question terms from Basque to English (see Section 2.2), and, finally, we retrieved the relevant passages for the translated query terms (see Section 2.3).

As we were interested in the performance of passage retrieval on its own, we did not carry out any answer validation, and we just chose the first passage returned by the passage retrieval module as the response. We did not leave any question unanswered.

For both tasks, the only difference between the submitted two runs is the use (or not) of the expansion in the passage retrieval module. That is, in the first run ("run 1" in Table 1), during the retrieval we only used the original words that were in the passage. In the second run ("run 2" in Table 1), apart from the original words, we also used the expanded words.

4 Results

Table 1 summarizes the results of our submitted runs, explained in Section 3.

Table 1. Results for submitted runs

submitted runs		#answered correctly	#answered incorrectly	c@1
English - English	run 1	211	289	0.42
	run 2	240	260	**0.48**
Basque - English	run 1	78	422	0.16
	run 2	91	409	**0.18**

The results show that the use of the expanded words (run 2) was effective for both tasks, improving the final result by 6 % in the monolingual task.

Figure 2 shows an example of a document expansion which was effective for answering the English question number 32: "*Into which plant may genes be introduced and not raise any doubts about unfavourable consequences for people's health?*"

In the second part of the example we can see some words that we obtained after applying the expansion process explained in Section 2.3.1 to the original passage showed in the example too. As we can see, there are some new words among the expanded words that are not in the original passage, such as *unfavourable* or *consequence*. Those two words were in the question we mentioned before (number 32). That could be why we answered that question correctly when using the expanded words (in run 2), but not when using the original words only.

original passage: *Whereas the Commission, having examined each of the objections raised in the light of Directive 90/220/EEC, the information submitted in the dossier and the opinion of the Scientific Committee on Plants, has reached the conclusion that there is no reason to believe that there will be any adverse effects on human health or the environment from the introduction into maize of the gene coding for phosphinotricine-acetyl-transferase and the truncated gene coding for beta-lactamase;*

some expanded words: *cistron factor gene coding cryptography secret_writing ... acetyl acetyl_group acetyl_radical ethanoyl_group ethanoyl_radical beta_lactamase penicillinase ... ec eec eu europe european_community european_economic_community european_union ... directive directing directional guiding citizens_committee committee environment environs surround surroundings corn ... maize zea_mays health wellness health adverse contrary homo human human_being man adverse inauspicious untoward gamboge ... unfavorable* ***unfavourable*** *... set_up expostulation objection remonstrance remonstration dissent protest believe light lightly belief feeling impression notion opinion ... reason reason_out argue jurisprudence law* ***consequence*** *effect event issue outcome result upshot ...*

Fig. 2. Example of a document expansion (doc_id: *jrc31998D0293-en.xml,* p_id*: 17)*

As expected, the best results were obtained in the monolingual task. With the intention of finding reasons to explain the significant performance drop in the bilingual run, we analyzed manually 100 query translations obtained in the query translation process of the 500 test queries, and detected several types of errors arising from both the question analysis process and from the query translation process. In the question analysis process, some lemmas were not correctly identified by the lemmatizer/tagger, and in other cases some entities were not returned by the lemmatizer/tagger causing us to lose important information for the subsequent translation and retrieval processes. In the query translation process, leaving aside the incorrect translation selections, the words appearing in the source questions were not exactly the ones that figured in many queries that had been correctly translated. In most cases this happened because the English source query word was not a translation candidate in the MRD. If we assume that the answers contain words that appear in the questions and therefore in the passage that we must return, this will negatively affect the final retrieval process.

5 Conclusions

The joint Elhuyar-Ixa team has presented a system which works on passage retrieval alone, without any question analysis and answer validation steps. Our English-English results show that good results can be achieved by means of this simple strategy. We experimented with applying semantic relatedness in order to expand passages prior to indexing, and the results are highly positive, especially for English-English. The performance drop in the Basque-English bilingual runs is significant, and is caused by the accumulation of errors in the analysis and translation of the query mentioned.

Acknowledgments

This work has been supported by KNOW (TIN2006-15049-C03-01), imFUTOURnet (IE08-233) and KYOTO (ICT-2007-211423). Arantxa Otegi's work is funded by a

PhD grant from the Basque Government. Part of this work was done while Arantxa Otegi was visiting Yahoo! Research Barcelona.

References

1. Ansa, O., Arregi, X., Otegi, A., Soraluze, A.: Ihardetsi: A Basque Question Answering System at QA@CLEF 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) Evaluating Systems for Multilingual and Multimodal Information Access. LNCS, vol. 5706, pp. 369–376. Springer, Heidelberg (2009) ISSN 0302-9743 ISBN 978-3-642-04446
2. Kim, S., Seo, H., Rim, H.: Information retrieval using word senses: Root sense tagging approach. In: Proceedings of SIGIR (2004)
3. Liu, S., Yu, C., Meng, W.: Word Sense Disambiguation in Queries. In: Proceedings of the 14th ACM Conference on Information and Knowledge Management, CIKM (2005)
4. Pérez-Agüera, J.R., Zaragoza, H.: Query Clauses and Term Independence. In: Evaluating Systems for Multilingual and Multimodal Information Access, 9th Workshop of the Cross-Language Evaluation Forum. LNCS, pp. 369–376. Springer, Heidelberg (2009) ISSN 0302-9743 ISBN 978-3-642-04446
5. Agirre, E., Soroa, A., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M.: A study on similarity and relatedness using distributional and WordNet-based approaches. In: Proceedings of the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL), Boulder, USA (2009)
6. Ezeiza, N., Aduriz, I., Alegria, I., Arriola, J.M., Urizar, R.: Combining Stochastic and Rule-Based Methods for Disambiguation in Agglutinative Languages. In: COLING-ACL, pp.380–384 (1998)
7. Alegria, I., Arregi, O., Balza, I., Ezeiza, N., Fernandez, I., Urizar, R.: Development of a Named Entity Recognizer for an Agglutinative Language. In: IJCNLP (2004)
8. Darwish, K., Oard, D.W.: Probabilistic structured Query Methods. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 338–344 (2003)
9. Pirkola, A.: The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 55–63 (1998)
10. Ballesteros, L., Bruce Croft, W.: Resolving Ambiguity for Cross-language Retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 64–71 (1998)
11. Monz, C., Dorr, B.J.: Iterative translation disambiguation for cross-language Information Retrieval. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 520–527 (2005)
12. Saralegi, X., López de Lacalle, M.: Comparing different approaches to treat Translation Ambiguity in CLIR: Structured Queries v. Target Co-occurrence Based Selection. In: 6th TIR Workshop (2009)
13. Boldi, P., Vigna, S.: MG4J at TREC 2005. In: Voorhees, E.M., Buckland, L.P. (eds.) The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings, Number SP 500-266 in Special Publications. NIST (2005), `http://mg4j.dsi.unimi.it/`

Are Passages Enough? The MIRACLE Team Participation in QA@CLEF2009

María Teresa Vicente-Díez, César de Pablo-Sánchez, Paloma Martínez, Julián Moreno Schneider, and Marta Garrote Salazar

Universidad Carlos III de Madrid, Avda. Universidad, 30,
28911 Leganés, Madrid, Spain
{tvicente,cdepablo,pmf,jmschnei,mgarrote}@inf.uc3m.es

Abstract. This paper summarizes the participation of the MIRACLE team in the Multilingual Question Answering Track at CLEF 2009. In this campaign, we took part in the monolingual Spanish task at ResPubliQA and submitted two runs. We have adapted our QA system to the new JRC-Acquis collection and the legal domain. We tested the use of answer filtering and ranking techniques against a baseline system using passage retrieval with no success. The run using question analysis and passage retrieval obtained a global accuracy of 0.33, while the addition of an answer filtering resulted in 0.29. We provide an analysis of the results for different questions types to investigate why it is difficult to leverage previous QA techniques. Another task of our work has been the application of temporal management to QA. Finally we include some discussion of the problems found with the new collection and the complexities of the domain.

1 Introduction

We describe the MIRACLE team participation in the ResPubliQA exercise in the Multilingual Question Answering Track at CLEF 2009. The MIRACLE team is a consortium formed by three universities from Madrid, (Universidad Politécnica de Madrid, Universidad Autónoma de Madrid and Universidad Carlos III de Madrid) and DAEDALUS, a small and medium size enterprise (SME). We submitted two runs for the Spanish monolingual subtask which summarize our attempts to adapt our QA system to the new requirements of the task.

The change in the application domain has been triggered by the use of the JRC-Acquis document collection [1], which is formed by European legislation translated in several EU languages. This fact raises the problem of dealing with legal language, which includes technical terminology and shows a more complex syntactic structure than news or academic language used in EFE and Wikipedia collections. Moreover, new information needs require the inclusion of questions asking for objectives, motivations, procedures, etc. in addition to the traditional factual and definition questions. The new types of questions often required longer answers and, therefore, the response of the system has been fixed again at the paragraph level. Nevertheless, it should be possible to take advantage of answer selection techniques developed in previous campaigns. This has been in fact one of the hypothesis we wanted to test. Unfortunately,

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 281–288, 2010.

our experiments in this line have not been successful and we have not found configurations that performed substantially better than our baseline. Another aspect of our work has focused on the use of temporal information in the process of QA. We report the results for different indexing configurations. Finally, a global objective was to enhance the capabilities of the QA system and advance towards an architecture that allows domain adaptation and multilingual processing.

The paper is structured as follows: section 2 describes the system architecture with special attention paid to the novelties introduced this year; section 3 presents the submitted runs and the analysis of the results. Finally, conclusions and future work are shown in section 4.

2 System Description

The system architecture is based on the approach taken by the MIRACLE QA system participating in CLEF 2008 [2] and consists in a pipeline which analyzes questions, retrieves documents and performs answer extraction based on linguistic and semantic information. A rule engine has been used in the Question Classification, Answer Filter, Timex Analyzer and Topic Detection modules. The left part of the rules are patterns that can refer to lexical, syntactic and/or semantic elements, whereas the right part are actions that add annotations like question types, entity classes or time normalizations. Figure 1 shows the architectural schema.

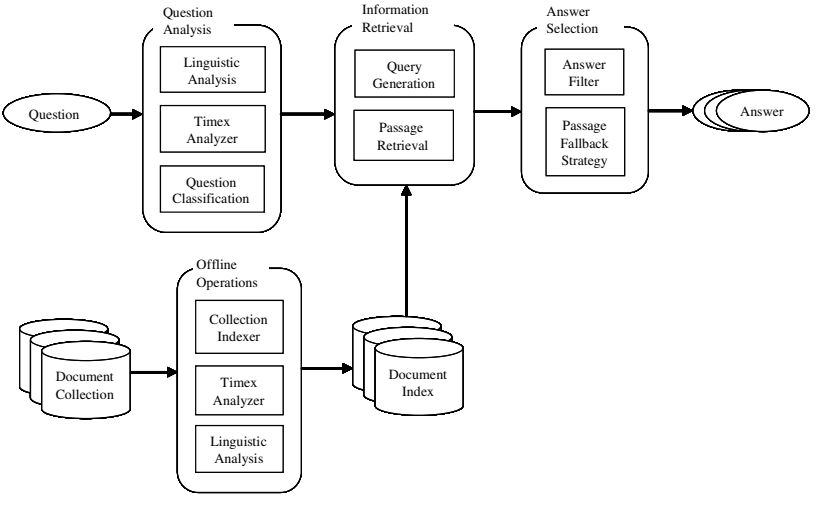


Fig. 1. MIRACLE 2009 system architecture

Some new modules have been included to carry out new experiments while others have been modified, extended or reorganized. The main changes are the following:

- Adding parsers for the new collections and supporting the indexing of passages.
- The evaluation procedure was modified to work with passages and a fallback strategy for passages was also included.
- New rules have been developed for Question Analysis and Answer Selection for the legal domain using the development set.
- Query Generation has been adapted to the domain, removing old heuristics.
- Temporal Management was added and integrated into indexing routines.
- New functionality for mining acronyms and adding them to Query Generation.
- The Ranking module was redesigned for modularity.

2.1 Indexes

Due to the change in the document collection, all IR indexes have been newly created using Lucene [3]. To accomplish the task of storing the relevant information as appropriately as needed, we have designed two different indexing units: *Document*, where all the information related to title, note and the text of the file is stored; and *Paragraph*, which stores each paragraph, title and the notes in a different unit. Lucene uses a length document normalization term in the retrieval score which was arguably of no help in the case of paragraph scoring because paragraphs are expected to have more uniform lengths. Both types of indexes, with and without length normalization, were tested.

In all our experiments the paragraph or passage index worked better than the document index. Besides, we also created different index types regarding the analysis, characterized by the linguistic analyzer used in each case: *Simple Index*, where the text analyzer used is a simple analyzer adapted for Spanish. It makes grammar based parsing, stems words using a snowball-generated stemmer, removes stop words, replaces accented characters and converts text into lower case. *Temporal Index*, which adds recognition and normalization of time expressions. These time expressions are normalized and included in the index.

2.2 Temporal Management

Some authors have defined the temporal question answering (TQA) as the specialization of the QA task in which questions denote temporality [4], as well as a means for providing short and focused answers to temporal information needs [5]. Previous work has already faced up to this problem [6], [7]. Temporal questions can be classified into 2 main categories according to the role of temporality in their resolution: *Temporally Restricted* (TR) questions are those containing some time restriction: *"What resolution was adopted by the Council on 10 October 1994?"*; and *Questions with a Timex Answer* (TA) are those whose target is a temporal expression or a date: *"When does the marketing year for cereals begin?"*

In this campaign, temporal management preserves the approach taken by our previous system [2]. This decision is based on later complementary work that was made in order to evaluate the QA system performance versus a baseline system without

temporal management capabilities [8]. The experiments showed that additional temporal information management can benefit the results.

Several adjustments were made in the temporal expressions recognition, resolution and normalization components to enhance their coverage on the new collections. The date of creation of each document is adopted as the reference date, needed to resolve the relative expressions that the collection could contain (for instance: "*yesterday",* or *"last week*"). This type of expressions need another point in time to be properly resolved, that is, to deduce their semantics. This point of reference could be a date taken from the context of the document but a simpler approach is to consider the date in which contents were created. In JRC-Acquis documents this information is provided by the "*date.created*" attribute. During question analysis process, queries, including those with temporal features, are classified, distinguishing between TR and TA queries. If a TA query is detected, it determines the granularity of the expected answer (complete date, only year, month, etc.). The answer selector is involved in two directions: in the case of TA queries, the module must favour a temporal answer, whereas if it manages TR queries, it applies extraction rules based on the temporal inference mechanism and demotes the candidates not fulfilling the temporal restrictions.

As a novelty, this year we have created more sophisticated indexes according to the paragraph retrieval approach of the competition. In some configurations, the normalized resolution of temporal expressions is included in the index instead of the expression itself [9]. The main objective is to assess the behavior of the QA system using different index configurations, focusing on the temporal queries.

2.3 Acronym Mining

Due to the nature of the collection, a large number of questions were expected to be expansion of acronyms, especially about organizations. On the other hand, the recall of the information retrieval step could be improved by including the acronym and their expansion in the query.

We implemented a simple offline procedure to mine acronyms by scanning the collection and searching for a pattern which introduces a new entity and provides their acronym between parentheses. Then, results are filtered in order to increase their precision. First, only those associations that occur at least twice in the corpus are considered. As parentheses often convey other relations like persons and their country of origin, another filter removed countries (*Spain*) and their acronyms (*ES*) from the list. Finally, some few frequent mistakes were manually removed and acronyms with more than one expansion were also checked. We indexed the acronyms and their expansions separately to be able to search by acronym or by expansion.

The acronym index is used in two different places in the QA system: during Query Generation, where it analyzes the question and adds search terms to the query; and in Answer Filtering, where it analyzes the text extracted from the paragraph to determine if that paragraph contains the acronym (or the expansion) and if so, identifies the paragraph as correct answer.

2.4 Answer Filter and Passage Fallback Strategy

This module, previously called Answer Extractor, processes the result list from the information retrieval module and selected chunks to form a possible candidate answer. In this campaign, the answer must be the complete text of a paragraph and, therefore, this year the module works as a filter which removes passages with no answers. The system has been adapted and new rules to detect acronyms, definitions, as expressed in the new corpora, and temporal questions have been developed.

The possibility of getting no answer from the Answer Filter led to the development of a module that simply creates answers from the retrieved documents. This module is called Passage Fallback Strategy. It takes the documents returned by the information retrieval module and generates an answer from every document.

2.5 Evaluation Module

Evaluation is a paramount part of the development process of the QA system. In order to develop and test the system, the English development test provided by CLEF organizers was translated to Spanish and a small gold-standard with answers was developed. Mean Reciprocal Rank (MRR) and Confidence Weighted Score (CWS) were consistently used to compare the output of the different configurations with the development gold standard. The output of different executions were manually inspected to complete the gold standard and to detect integration problems.

3 Experiments and Results

We submitted two runs for the monolingual Spanish task. They correspond to the configurations of the system that yielded best results during our development using the translated question set.

The first configuration consisted on a version of the system that includes modules for Question Analysis and Information Retrieval together with a number of Offline Operations that perform the linguistic analysis of the collection and originate the indexes. Moreover the management of time expressions (Timex Analysis) was eliminated both in the collection and in the query processing looking for avoiding ambiguity in the semantics of numerical expressions. The second configuration was based on the addition of an Answer Selection strategy to the first design (Figure 1).

We called this runs *mira091eses* and *mira092eses*, each one corresponding to one of the previous configurations as follows:

- Baseline (BL): *mira091eses*. The system is based on passage retrieval using the Simple Index. Question Analysis is performed to generate queries and the acronym expansion is used.
- Baseline + Answer Filter (BL+AF): *mira092eses*. The Answer Filter and the Passage Fallback Strategy modules are added after the previous passage retrieval.

A number of additional configurations were also tested, but no improvements over the baseline were found. In fact, most of the additions seemed to produce worse results. We considered different functions for answer and passage ranking. Different passage

length normalization strategies were also applied to the indexes. Finally, a great deal of effort was devoted to the management of temporal expressions; more detailed experiments are presented below.

Evaluation figures are detailed in Table 1. Answer accuracy (*Acc*) has been calculated as the ratio of questions correctly answered (*Right*) to the total number of questions. Only the first candidate answer is considered, rejecting the rest of possibilities.

Table 1. Results for submitted runs

Name	Rigth	Wrong	Unansw. Right Candidate	Unansw. Wrong Candidate	Unansw. Empty Candidate	Acc.	Correctly discarded	c@1 measure
mira091eses	161	339	0	0	0	**0.32**	0	0.32
mira092eses	147	352	0	0	1	**0.29**	0	0.29

The results on the CLEF09 test set show similar conclusions to those obtained during our development process: the baseline system using passage retrieval is hard to beat; our second run provides lower accuracy. As in the case of our development experiments, there are changes for individual answers of certain questions, but the overall effect is not positive.

We have decided to carry a class based analysis in order to understand the causes behind our unfruitful efforts. We have manually annotated the test set and grouped questions into 6 main types (see Table 2). Contrary to our expectations, the performance of the second submitted run is also worse for the factual and definition questions. As we had considered these questions types in previous evaluations, we expected to have better coverage. Similar behavior has been observed across answer types for factual questions, being the class of temporal questions the only where a more complex configuration really improves.

Our analysis of the errors show that further work is needed to be able to cope with the complexities of the domain. For example, questions are in general more complex and include domain specific terminology that our question analysis rules do not handle correctly. The process of finding the focus of the question, which is crucial for question classification, is specially error prone. Answer Selection needs also further adaptation to the domain for factual questions as the typology of Named Entities (NE) and generalized NE has not wide coverage. On the other hand, being the first time that the legal domain was used, there was not any previous knowledge about how good would be the performance using existing rules of the system in a new context, without a gold standard to suggest some tuning actions.

Problems with definitions are rooted more deeply and probably require the use of different specialized retrieval strategies. This year evidence along with previous experiments seems to support that definitions depend deeply on the stylistics of the domain. Finally, new question types would require further study of techniques that help to improve the classification of passages as bearing procedures, objectives, etc.

Table 2. An analysis of runs by question type

Question Type	TOTAL (test set)	mira091eses Right	mira091eses Acc	mira092eses Right	mira092eses Acc
FACTUAL	123	54	0.44	48	0.39
PROCEDURE	76	22	0.28	15	0.20
CAUSE	102	43	0.42	44	0.43
REQUIREMENT	16	5	0.31	5	0.31
DEFINITION	106	16	0.16	12	0.11
OBJECTIVE	77	21	0.27	23	0.30
ALL	500	161	0.32	147	0.29

Evaluation of Temporal Questions

We extracted the temporal questions from the whole corpus: 46 out of 500 queries denote temporal information, which means a 9.20% over the total. 24 of them are TR questions, whereas TA queries are 22 (4.80% and 4.40% out of the total, respectively). This subset has been studied, evaluating the correctness of the answers by two different configurations of the system. The results are presented in Table 3.

Table 3. Results for temporal questions in the submitted runs and additional configurations

Name	Temporal Questions (TR + TA)	Temporally Restricted (TR)	Timex Answer (TA)
BL (mira091eses)	0.43	0.42	0.45
BL-AF (mira092eses)	0.48	0.37	0.59
DA-BL (non-submitted configuration 1)	0.28	0.21	0.36
DA-BL-AF (non-submitted configuration 2)	0.37	0.21	0.54

Better figures are obtained by the set of TQ in both runs. There is no significant difference between TA and TR queries in the first run, while in the second one they achieve a difference of 22%. In our opinion, the second configuration enhances precision for TA queries, whereas for TR queries, temporal restrictions introduce noise that the system is not able to solve.

Non-submitted runs present similar configurations to the submitted ones, but adopting a different index generation and question analysis strategies. The approach consisted of the inclusion of normalized temporal expressions into the index, as well as into the question analysis process, aiming to increase recall. We tested the performance over the total corpus, but worse results were achieved even if the study is restricted to temporal questions. Results show no improvement regarding the submitted runs. Performance difference between TA and TR queries remains stable, since the system has a better response to questions without temporal restrictions. Once the results were analyzed, we consider incorrect our initial assumption of extracting the reference date from the "*date.created*" attribute of the documents. This hypothesis could be partially the cause of erroneously resolving almost all relative dates. This is due to the fact that we assumed that this attribute was the date of creation of the document, whereas actually it refers to the date of publication of the collection, without providing any significant context information. Besides, lost of accuracy can be due to the lack of a more sophisticated inference mechanism at the time of retrieval, capable of reasoning with different granularities of normalized dates.

4 Conclusion and Future Work

From our point of view, the new ResPubliQA exercise is a challenge for QA systems in two main senses: linguistic domain adaptation and multilingualism. This year our efforts have focused on the first problem, adapting the previous system to the new collection. However, our experiments show that a system mainly based on passage retrieval performs quite well. Baseline passage retrieval results provided by the organizers [10] also support this argument. We are carrying out further experiments to find how answer selection could help for ResPubliQA questions, as well as the differences between passage retrieval alternatives. Regarding our task on temporal reasoning applied to QA, we will explore how question temporal constraints can be integrated at other steps in the process. We expect to compare the effectiveness of temporal reasoning as constraints for filtering answers and for the purpose of re-ranking. Finally, further work in the general architecture of the QA system is planned regarding three areas: separation of domain knowledge from general techniques, adding different languages to the system and effective evaluation.

Acknowledgements. This work has been partially supported by the Research Network MAVIR (S-0505/TIC/000267) and by the project BRAVO (TIN2007-67407-C3-01).

References

1. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., Varga, D.: The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, Italy (2006)
2. Martínez-González, A., de Pablo-Sánchez, C., Polo-Bayo, C., Vicente-Díez, M.T., Martinez-Fernández, P., Martínez-Fernández, J.L.: The MIRACLE Team at the CLEF 2008 Multilingual Question Answering Track. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 409–420. Springer, Heidelberg (2009)
3. Apache Lucene project. The Apache Software Foundation, `http://lucene.apache.org`
4. Saquete, E.: Resolución de Información Temporal y su Aplicación a la Búsqueda de Respuestas. Thesis in Computer Science, Universidad de Alicante (2005)
5. De Rijke, et al.: Inference for temporal question answering Project (2004-2007)
6. Clark, C., Moldovan, D.: Temporally Relevant Answer Selection. In: Proceedings of the 2005 International Conference on Intelligence Analysis (2005)
7. Saquete, E., Martínez-Barco, P., Muñoz, R., Vicedo, J.: Splitting Complex Temporal Questions for Question Answering Systems. In: Proceedings of the ACL 2004 Conference, Barcelona, Spain (2004)
8. Vicente-Díez, M.T., y Martínez, P.: Aplicación de técnicas de extracción de información temporal a los sistemas de búsqueda de respuestas. Procesamiento del lenguaje natural (42), 25–30 (2009)
9. Vicente-Díez, M.T., Martínez, P.: Temporal Semantics Extraction for Improving Web Search. 8th International Workshop on Web Semantics (WebS 2009). In: Tajoa, A.M., Wagner, R.R. (eds.) Proceedings of the 20th International Workshop on Database and Expert Systems Applications, DEXA 2009, pp. 69–73. IEEE Press, Los Alamitos (2009)
10. Pérez, J., Garrido, G., Rodrigo, A., Araujo, L., Peñas, A.: Information Retrieval Baselines for the ResPubliQA task. In: CLEF 2009 Working Notes (2009)

The LIMSI Participation in the QAst 2009 Track: Experimenting on Answer Scoring

Guillaume Bernard[1], Sophie Rosset[1], Olivier Galibert[2], Gilles Adda[1], and Eric Bilinski[1]

[1] Spoken Language Processing Group, LIMSI-CNRS
[2] Laboratoire National de Métrologie et d'Essai (LNE)
{gbernard,rosset,gadda,bilinski}@limsi.fr, olivier.galibert@lne.fr

Abstract. We present in this paper the three LIMSI question-answering systems on speech transcripts which participated to the QAst 2009 evaluation. These systems are based on a complete and multi-level analysis of both queries and documents. These systems use an automatically generated research descriptor. A score based on those descriptors is used to select documents and snippets. Three different methods are tried to extract and score candidate answers, and we present in particular a tree transformation based ranking method. We participated to all the tasks and submitted 30 runs (for 24 sub-tasks). The evaluation results for manual transcripts range from 27% to 36% for accuracy depending on the task and from 20% to 29% for automatic transcripts.

1 Introduction

The Question Answering on Speech Transcripts track of the QA@CLEF task provides an opportunity to evaluate the specificity of speech transcriptions. In this paper, we present the work carried out on the QA system developed at LIMSI for the QAst evaluation. We especially describe an answer re-ranking method used in this system.

For the QAst 2009 evaluation [8], 3 main tasks are defined:

- T1, QA in English European Parliament Plenary sessions
- T2, QA in Spanish European Parliament Plenary sessions
- T3, QA in French Broadcast News

For each of the tasks, four versions of the data collection were provided, consisting of one manual transcriptions and three different automatic transcription. Two different sets of questions were provided, one consisting of written questions and the other of manually transcribed semi-spontaneous oral questions [8]. In total a minimum of 8 runs were expected per task, for a total of 24. LIMSI participated to the three tasks. Three systems were tested. Their main architecture is identical and they differ only in the answer scoring method:

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 289–296, 2010.

– Distance-based answer scoring (primary method)
– Answer scoring through bayesian modeling
– Tree transformation-based answer re-ranking

The first method is used on all three tasks, the second one is used on the T1 and T2 tasks and the third one on the T3 task.

The section 2 presents the common architecture and the three answer scoring methods. The section 3 is split into two parts: the description of the training and development data (section 3.1), and the results of the three systems on the development and test data (section 3.2). We compare these results to those obtained in the QAst 2008 evaluation.

2 The LIMSI QA Systems

The common architecture is identical to the systems used in the previous evaluations and is fully described in [6].

The same complete and multilevel analysis is carried out on both queries and documents. To do so, the query and the documents (which may come from different modalities – text, manual transcripts, automatic transcripts) are transformed into a common representation. This normalization process converts *raw* texts to a form where words and numbers are unambiguously delimited, punctuation is separated from words, and the text is split into sentence-like segments. Case and punctuation are reconstructed using a fully cased, punctuated four-gram language model [3] applied to a word graph covering all the possible variants (all possible punctuations permitted between words, all possible word cases). The general objective of this analysis is to find the bits of information that may be of use for search and extraction, called *pertinent information chunks*. These can be of different categories: named entities, linguistic entities (e.g., verbs, prepositions), or specific entities (e.g., scores). All words that do not fall into such chunks are automatically grouped into chunks via a longest-match strategy. The full analysis comprises some 100 steps and takes roughly 4 ms on a typical user or document sentence. The analysis identifies about 300 different types of entities. The analysis is hierarchical, resulting in a set of trees. Both answers and important elements of the questions are supposed to be annotated as one of these entities.

The first step of QA system itself is to build a search descriptor (SD) that contains the important elements of the question, and the possible answer types with associated weights. Some elements are marked as *critical*, which makes them mandatory in future steps, while others are *secondary*. The element extraction and weighting is based on an empirical classification of the element types in importance levels. Answer types are predicted through rules based on combinations of elements of the question.

Documents are selected using this SD. Each element of the document is scored with the geometric mean of the number of occurrences of all the SD elements that appear in it, and sorted by score, keeping the n-best. Snippets are extracted

from the document using fixed-size windows and scored using the geometrical mean of the number of occurrences of all the SD elements that appear in the snippet, smoothed by the document score.

2.1 Distance-Based Answer Scoring

In each snippet, all the elements whose type is one of the predicted possible answer types are candidate answers. A score $S(r)$ is associated to each candidate answer r.

This score is the sum of the the distances between itself and the elements of the SD, each elevated to the power $-\alpha$, ponderated by the element weights. That score is smoothed with the snippet score through a δ-ponderated geometric mean. All the scores for the different instances of the same element are added together, and in order to compensate for the differencing natural frequencies of the entities in the documents the final score is divided by the occurence count in all the documents and in all the examined snippets, each elevated to the power β and γ respectively. The entities with the best scores then win.

2.2 Answer Scoring through Bayesian Modeling

We tried a preliminary method of answer scoring built upon a bayesian modeling of the process of estimating the quality of an answer candidate. This approach relies on multiple elementary models including element co-occurrence probabilities, question element appearance probability in the context of a correct answer and out of context answer probability. The models parameters are either estimated on the documents or are set empirically. This is a very preliminary work.

2.3 Tree Transformation-Based Answer Re-ranking

Our second approach for the T3 task is built upon the results of the primary system. We stated that the method for finding and extracting the best answer to a given question in 2.1 is based on redundancy and distances between candidate answers and elements of the question. While this approach gives good results, it also has some limitations. Mainly, it does not take into account the structure of the snippet and the relations between the different critical elements detected.

Relations between the elements of the text fragments are needed to represent the information stated in the documents and the questions. However, most of the systems use complex syntactic representations which are not adapted to handle oral fragments [4]. Some systems [7,5] also show that it is possible to identify local syntactic and semantic relations by using a segmentation of the documents into segments (chunks) and then detecting the relations between these segments.

From these conclusions, we defined a re-ranking method which computes a score for each of the answers to a question. That method takes as input the question tagged by the analysis module, the answers found by the answer extraction module, and the best snippets associated to each answer. The analysis

trees of the question and the snippets are segmented into chunks, and relations are added between these chunks.

For each evaluated answer, the method compares the structure of the question with the snippet of the answer. The system tries to match the structure of the question by moving the chunks of the snippets with similar elements. The relations are used in these moves to compute the final score of the answer.

The relations are built in two steps. First the text is segmented in syntactically-motivated chunks with five possible types : verbal chunks (VC), temporal chunks (TC), spatial/geographical chunks (SC), question markers (QMC), and finally general chunks (GC) covering everything else. That annotation is done using a standard CRF approach based on words and POS annotations, using separate models for documents and for questions. Once the text is segmented the local relations are then established, belonging to four simple types: noun-modifier relations, verb-member relations, temporal modifier relations and spacial modifier relations. That relation annotation is done incrementally using a set of contextual rules. Here is an exemple of the results given by these two steps:

[GC] The Ebola virus [/GC] [VC] was identified [/VC] [TC] in 1976 [/TC].

1. Verb-member: *was identified* - *The Ebola virus*
2. Verb-member: *was identified* - *in 1976*
3. Temporal modifier: *in 1976* - *The Ebola virus*
4. Temporal modifier: *in 1976* - *was identified*

Once the relations are established, we then can transform the snippet into the question. As a preliminary, the question marker chunk of the question is replaced by the to-be-reranked answer. Then the hybrid heuristic-search method we use starts by detecting *anchor points* between the question and the snippet of the answer, which are pairs of chunks identical through lemmatization, synonymy and/or morphological derivations. Using these anchor points, the system starts by using a *substitution* operation to transform the chunks of the snippet into those of the questions. It also detects if the substituated chunk in the snippet has a relation with the chunk of the evaluated answer. If not, an *attachement* operation is applied. Each operation type has a cost, the substitution one depending of the transformations applied, and the attachement cost depending on the change in relations as annotated previously between the perturbed chunks. The sequence of operations with the lowest total cost to transform all of the tranformable elements is searched for using a breadth-first technique. A cost for the suppression of the extra chunks and addition of the missing ones is added by using *suppression* and *insertion* operations. The final cost, the distance between the question/answer pair and the snippet, is used to rerank the proposed answers. A more detailed explanation of this approach can be found in [2].

3 Evaluation

3.1 Tuning and Development Data

To tune our base systems, we used the development corpus of the QAst 2009 evaluation and the data of the QAst 2008 evaluation [6].

The tree transformation method was evaluated on the training corpus. The approach obtains better results when the search descriptor contains at least five elements (see Table 1). That is why we decided to apply this new method on these conditions only.

The bayesian modeling parameters estimation is done only on the documents data and does not use question/answers pairs. It is probably one of its numerous flaws, explaining its relatively bad results.

Table 1. Results on the development data classified by number of search elements in the search descriptor, with #E being the number of search elements. LIMSI2-T3 is the re-ranking method and LIMSI1 the distance-based method.

	LIMSI2-T3			LIMSI1			
#E.	MRR	Acc	#Correct	MRR	Acc	#Correct	#Questions
1	0.62	48.6	53	0.71	67.0	73	109
2	0.56	42.2	73	0.66	61.8	107	173
3	0.74	67.4	145	0.79	77.9	166	215
4	0.72	65.5	74	0.79	77.9	88	113
5	**0.73**	**65.5**	38	0.71	60.3	35	58
6	**0.85**	**85.7**	18	0.81	76.0	16	21
7	0.60	60.0	3	0.60	60.0	3	5

3.2 Results

The results for the three tasks on manual transcribed data are presented in table 2, with all the question types evaluated. In every case the LIMSI1 system is the base system. LIMSI2-T1 and T2 are the bayesian system, and LIMSI2-T3 is the reranking system. As stated before, the reranking system in the T3 task is not used on all the questions, but only the questions with five or more search elements.

The results obtained for the three tasks on automatically transcribed data are presented in tables 3 to 5. With the automatic transcripts, only the base system is used. The Δ in each table shows the variation on accuracy between the manual and automatic transcriptions results.

3.3 Analysis of the Results

Tables 2 shows a great loss between the recall and the accuracy of our systems. The base system gives a bad answer at first rank on half of the questions with the good answer in the candidates answers, and it is worse for the bayesian system on the T1 and T2 tasks. The reranking system on the T3 task gives almost the same results that the base system, but it was applied only on to a small set of questions. We can see that there are almost no differences between written and spoken questions.

Table 2. Results for English EPPS, Spanish EPPS and French Broadcast news on manual transcripts

System	Questions	English			Spanish			French		
		MRR	Acc	Recall	MRR	Acc	Recall	MRR	Acc	Recall
LIMSI1	Written	0.36	27%	53%	0.45	36.0%	61%	0.39	28.0%	60%
	Spoken	0.33	23%	45%	0.45	36.0%	62%	0.39	28.0%	59%
LIMSI2	Written	0.32	23%	45%	0.34	24.0%	49%	0.38	27.0%	60%
	Spoken	0.27	19%	41%	0.34	24.0%	49%	0.39	28.0%	59%

Table 3. Results for task T1, English EPPS, automatic transcripts (75 factual questions and 25 definitional ones). The Δ measures the difference between the manual and automatic transcriptions on accuracy.

ASR	System	Questions	English			
			MRR	Acc	Recall	Δ
ASR_A	LIMSI1	Written	0.31	26.0%	42%	-1
10.6%		Spoken	0.30	25.0%	41%	-2
ASR_B	LIMSI1	Written	0.25	21.0%	32%	-2
14.0%		Spoken	0.25	21.0%	33%	-4
ASR_C	LIMSI1	Written	0.24	21.0%	31%	-4
24.1%		Spoken	0.24	20.0%	33%	-5

Table 4. Results for task T2, Spanish EPPS, automatic transcripts (44 factual questions and 56 definitional ones). The Δ measures the difference between the manual and automatic transcriptions on accuracy.

ASR	System	Questions	Spanish			
			MRR	Acc	Recall	Δ
ASR_A	LIMSI1	Written	0.32	27.0%	42%	-9
11.5%		Spoken	0.31	26.0%	41%	-10
ASR_B	LIMSI1	Written	0.29	25.0%	37%	-11
12.7%		Spoken	0.29	25.0%	37%	-11
ASR_C	LIMSI1	Written	0.28	23.0%	37%	-13
13.7%		Spoken	0.28	24.0%	37%	-12

The bayesian system is a preliminary version that gives interesting results but has shown that some of our modelization hypothesis were incorrect. Measurements on real data will help us enhance it. For the reranking system only ten questions of the written question corpus has enough search elements to be considered. Six of them did not have the correct answer within the candidates answers and one was a NIL question. Of the remaining three, one was answered correctly by both systems, one was answered correctly by the base system but not the reranking one. And the correct answer for the last question was not

Table 5. Results for the T3 task, French Broadcast News, manual transcripts (68 factual questions and 32 definitional ones). The Δ measures the difference between the manual and automatic transcriptions on accuracy.

ASR	System	Questions	French			
			MRR	Acc	Recall	Δ
ASR_A	LIMSI1	Written	0.37	29.0%	52%	+1
11.0%		Spoken	0.37	29.0%	50%	+1
ASR_B	LIMSI1	Written	0.32	27.0%	40%	-1
23.9%		Spoken	0.30	25.0%	38%	-3
ASR_C	LIMSI1	Written	0.28	23.0%	38%	-5
35.4%		Spoken	0.27	22.0%	35%	-6

found by either of the systems. As such, the reranking system still needs work to improve it.

For the results obtained on the three different automatic speech transcription, as shown in tables 3 to 5, we can see that they are as expected lower than the results of the manual transcriptions. For English and French, the loss are roughly the same. On Spanish thought, we see a strong decrease in the results.

Table 6. Results obtained by the LIMSI on the QAst 2009 evaluation

		T1		T2		T3	
Sub-Task	**Question**	**Acc.**	**Best**	**Acc.**	**Best**	**Acc.**	**Best**
Manual	Written	27.0%	28.0%	36.0%	-	28.0%	-
	Spoken	23.0%	26.0%	36.0%	-	28.0%	-
ASR_A	Written	26.0%	-	27.0%	-	29.0%	-
	Spoken	25.0%	-	26.0%	-	29.0%	-
ASR_B	Written	21.0%	-	25.0%	-	27.0%	-
	Spoken	21.0%	-	25.0%	-	25.0%	-
ASR_C	Written	21.0%	25.0%	23.0%	-	23.0%	-
	Spoken	20.0%	25.0%	24.0%	-	22.0%	-

Table 6 shows the results obtained by the LIMSI on each task. We also show the best results of all the participants' systems in column *Best* for each task. Except on the T1 Manual and the T1 ASR_A, the LIMSI obtained the best results. It should be noted that we were the only participants in the T3 (french) task. These results confirm that our approach regarding spoken data processing is well adapted and robust.

We observed a strong loss between QAst 2008 results [6] and QAst 2009 results, especially on the English and French tasks. It seems that the new methodology used to build the questions corpus is a probable cause for these results. We are currently studying the impact of this new methodology, in particular by evaluating the mean distance between the elements of each question and the answer to this question. The first results of this study can be found in [1].

4 Conclusion

In this paper, we presented the LIMSI question-answering systems on speech transcripts which participated to the QAst 2009 evaluation. These systems obtained state-of-the-art results on the different tasks and languages and the accuracy ranged from 27% for English to 36% for Spanish data). The results of the T1 ans T3 systems show a significant loss of results compared to the 2008 evaluation (6% for T1 and 17% for T3 in accuracy) inspite of the improvements of the systems. It can probably be explained by the new methodology used to build the questions corpus. A deeper analysis is ongoing to understand the origins of the loss. The base system still obtains the best results. The results of the other systems are currently analysed to improve the two approach. In particular, we are evaluating application cases for the tree transformation method.

Acknowledgments

This work has been partially financed by OSEO under the Quaero program.

References

1. Bernard, G., Rosset, S., Adda-Decker, M., Galibert, O.: A question–answer distance measure to investigate QA system progress. In: LREC 2010 (2010)
2. Bernard, G., Rosset, S., Galibert, O., Bilinski, E., Adda, G.: The limsi participation to the qast 2009 track. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece (October 2009)
3. Déchelotte, D., Schwenk, H., Adda, G., Gauvain, J.-L.: Improved machine translation of speech-to-text outputs. In: Interspeech 2007, Antwerp, Belgium (2007)
4. Paroubek, P., Vilnat, A., Grau, B., Ayache, C.: Easy, evaluation of parsers of french: what are the results. In: European Language Resources Association (ELRA) (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), pp. 2480–2486, Marrakech, Morocco (2008)
5. Pradhan, S., Ward, W., Hacioglu, K., Martin, J.H., Jurafski, D.: Semantic role labeling using different syntactic views. In: Proceedings of the 43rd Annual Meeting of the ACL, Ann Arbor, USA, pp. 581–588 (2005)
6. Rosset, S., Galibert, O., Bernard, G., Bilinski, E., Adda, G.: The limsi participation to the qast track. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 480–487. Springer, Heidelberg (2009)
7. Sakai, T., Saito, Y., Ichimura, Y., Koyama, M., Kokubu, T., Manabe, T.: Askmi: A japanese question answering system based on semantic role analysis. In: Proceedings of RIAO 2004, Avignon (2004)
8. Turmo, J., Comas, P., Rosset, S., Galibert, O., Moreau, N., Mostefa, D., Rosso, P., Buscaldi, D.: Overview of qast 2009 - question answering on speech transcriptions. In: CLEF 2009 Workshop, Greece, Corfu (2009) (to appear)

Robust Question Answering for Speech Transcripts: UPC Experience in QAst 2009

Pere R. Comas and Jordi Turmo

TALP Research Center, Technical University of Catalonia (UPC)
pcomas@lsi.upc.edu, turmo@lsi.upc.edu

Abstract. This paper describes the participation of the Technical University of Catalonia in the CLEF 2009 Question Answering on Speech Transcripts track. We have participated in the English and Spanish scenarios of QAST. For both manual and automatic transcripts we have used a robust factual Question Answering that uses minimal syntactic information. We have also developed a NERC designed to handle automatic transcripts. We perform a detailed analysis of our results and draw conclusions relating QA performance to word error rate and the difference between written and spoken questions.

1 Introduction

The CLEF 2009 Question Answering on Speech Transcripts (QAST) track [7] consists of four Question Answering (QA) tasks for three different languages: T1 English, T2 Spanish and T3 French. Task m is QA in manual transcripts of recorded European Parliament Plenary Sessions (EPPS). Tasks a, b, and c, use three different transcripts of the recorded audio using three Automatic Speech Recognizers (ASR). These transcriptions have an increasing percentage of errors. There are two sets of questions for each language: set B contains oral questions spontaneously asked by several human speakers, while set A consists of grammatically corrected transcriptions of the questions in set B. The questions are divided in two sets of development (50 questions) and test (100 questions). Given the languages, questions and transcripts, there is a total of 24 possible scenarios in the QAST evaluation. For example, we will refer as T2B-a to the scenario taking the best automatic transcripts of the Spanish EPPS using spontaneous questions. The automatic transcripts have different levels of word error rate (WER). WERs for T1 are 10.6%, 14%, and 24.1%. For T2 WERs are 11.5%, 12.7% and 13.7%. Figure 1 shows a text sample extracted from the T1 corpus.

This paper summarizes our methods and results in QAST. We have participated in scenarios T1 and T2 with all transcripts and question sets. Our QA system is based on our previous work in [1] and [6]. We have used the same system architecture for all the tasks, having interchangeable language–dependant parts and different passage retrieval algorithms for automatic transcripts.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 297–304, 2010.

T1-*m*: *"Abidjan is going going the way of Kinshasa Kinshasa which was of course a country in the past with skyscrapers and boulevards and now a country a city in ruins"*

T1-*a*: "average down is going to go in the way of Kinshasa other at Kinshasa which was of course a country in the past of skyscrapers and poorer parts and our country as a city in ruins"

Fig. 1. Sample of manual and automatic transcripts

2 Overview of the System Architecture

The architecture of our QA system follows a commonly-used schema which splits the process of QA into three phases performed in a sequential pipeline: Question Processing (QP), Passage Retrieval (PR), and Answer Extraction (AE). This QA system is designed to answer to factoid questions, those whose answer is a named entity (NE).

2.1 Question Processing and Classification (QC)

The main goal of this component is to detect the type of the expected answer. We currently recognize the 53 open-domain answer types from [4]. The answer types are extracted using a multi-class Perceptron classifier and a rich set of lexical, semantic and syntactic features. This classifier obtains an accuracy of 88% on the corpus of [4]. Additionally, the QP component extracts and ranks relevant keywords from the question

For scenario T2, he have developed an Spanish question classifier using human–translated questions from the corpus of [4] following the same machine learning approach. This classifier obtains an accuracy of 74%.

2.2 Passage Retrieval (PR)

This component retrieves a set of relevant passages from the document collection, given the previously extracted question keywords. The PR algorithm uses a query relaxation procedure that iteratively adjusts the number of keywords used for retrieval and their proximity until the quality of the recovered information is satisfactory (see [6]). In each iteration a Document Retrieval application (IR engine) fetches the documents relevant to the current query and a subsequent passage construction module builds passages as segments where two consecutive keyword occurrences are separated by at most t words.

When dealing with automatic transcripts, the ASR may introduce incorrect words that were not actualy uttered to the transcription. This may create problems to the IR engine sice some relevant passages may be impossible to find and some irrelevant passages can contain unrelated terms.

To overcome such drawbacks, we created an IR engine relying on phonetic similarity for the automatic transcripts. This tool is called PHAST (after PHonetic Alignment Search Tool) and uses pattern matching algorithms to search for small sequences of phones (the keywords) into a larger sequence (the documents) using a measure of sound similarity. Then the PR algorithm may be applied to the words found with PHAST. A detailed description of PHAST can be found in [2].

2.3 Answer Extraction (AE)

This module extracts the exact answer from the retrieved passages. First, answer candidates are identified as the set of named entities (NEs) that occur in these passages and have the same type as the answer type detected by QP (e.g. for the question *"When were biometric data included in passports?"* all retrieved entities of types date or time are identified as candidate answers). Then, these candidates are ranked using a scoring function based on a set of heuristics that measure keyword distance and density [5]. These heuristic measures use approximated matching for AE in automatic transcripts as shown in the passage retrieval module from the previous section. The same measure is used for T1 and T2.

3 Named Entity Recognition and Classification (NERC)

As described before, we extract candidate answers from the NEs that occur in the passages retrieved by the PR component. We detail below the strategies used for NERC in both manual and automatic transcripts.

We have taken a machine learning approach to this problem. First we apply learning at word level to identify NE candidates using a BIO tagging scheme. Then these candidates are classified into NE categories. Each function is modeled with *voted perceptron* [3]. As learning data we have manually labeled the NEs that occur in the QAST corpora T1 and T2 with their types (i.e. date, location, number, organization, person and time).

Our NERC uses a rich set of lexical and syntactic features which are standard to state-of-the-art NERCs. This features include: words, lemmas, POS tags, word affixes, flags regarding presence of numerals and capitalization, use of gazetteers, and n-grams of this features within a certain window of words. New features specially designed for automatic transcripts have been added to the sets *a*, *b* and *c*. These features use phonetic transcription of the words:

- Prefixes and suffixes of phones.
- Phonetic similarity with words in the gazetteer.
- A clustering of the transcriptions of the words has been done by grouping words with similar pronunciation. This clustering reduces the sparseness of the word–based features by mapping the words in several smaller subsets of different coarseness.

– n-grams of the forementioned clusters. This captures the possibility of splitting and merging adjacent words in order to overcome some ASR recognition errors.

The addition of these phonetic features improves the results by no more than 2 points of $F_{\beta=1}$ score in datasets *a*, *b* and *c*.

Given that there is no specific datasets for development, and we do not have more automatic transcripts of EPPS data, it is not possible to train our NERC on a dataset other than the test set. Therefore we have relabeled both corpora through a process of cross–validation. Both corpora have been randomly split in 5 segments, a NERC model has been learned for all subsets of 4 segments and the remaining segment has been labeled using this model. Thus we can train our NERC with documents from the same domain but test it on unseen data.

Table 1. NERC performance

T1: English					T2: Spanish				
Set	WER	Precision	Recall	$F_{\beta=1}$	Set	WER	Precision	Recall	$F_{\beta=1}$
m	∼ 0%	70.63%	68.19%	69.39	*m*	∼ 0%	76.11%	71.19%	73.57
a	10.6%	63.57%	55.26%	59.13	*a*	11.5%	72.40%	62.03%	66.81
b	14%	61.51%	52.28%	56.52	*b*	12.7%	64.33%	55.95%	59.85
c	24.1%	58.62%	43.92%	50.22	*c*	13.7%	67.61%	55.60%	61.02

Table 3 shows the overall results of our NERC for the four datasets of both T1 and T2. As the transcript WER increases, the scores consequently drop.[1]

4 Experimental Results

UPC participated in 2 of the 3 scenarios, the English (T1) and Spanish (T2) ones. We submitted two runs for each task, run number 1 uses the standard NERC described in Section 3 and run number 2 uses the hand–annotated NEs. Rows marked with † denote post–QAST results. These runs are discussed later. Each scenario included 100 test questions, from which 20 do not have an answer in the corpora (these are *nil* questions). In T1 75 question are factoids for 44 in T2. Our QA system is designed to answer only factual questions, therefore the our experimental analysis will refer only to factual questions.

We report two measures: (a) TOPk, which assigns to a question a score of 1 only if the system provided a correct answer in the top k returned; and (b) Mean Reciprocal Rank (MRR), which assigns to a question a score of $1/k$, where k is the position of the correct answer, or 0 if no correct answer is found. The official evaluation of QAST 2009 uses TOP1 and TOP5 measures [7]. An answer is considered correct by the human evaluators if it contains the complete

[1] Consider manual transcripts *m* having a WER of almost 0%.

Table 2. Overall factoid results for our sixteen English runs

Task, System	#Q	MRR	TOP1	TOP5	Task, System	#Q	MRR	TOP1	TOP5
T1A-*m* 1	75	0.27	14	32	T1A-*m* 2	75	0.31	17	35
T1B-*m* 1	75	0.15	7	19	T1B-*m* 2	75	0.15	7	18
T1B-*m* 1†	75	0.24	11	28	T1B-*m* 2†	75	0.24	13	27
T1A-*a* 1	75	0.27	14	29	T1A-*a* 2	75	0.26	14	30
T1B-*a* 1	75	0.08	4	11	T1B-*a* 2	75	0.09	4	12
T1B-*a* 1†	75	0.24	13	26	T1B-*a* 2†	75	0.22	10	28
T1A-*b* 1	75	0.24	13	26	T1A-*b* 2	75	0.26	15	29
T1B-*b* 1	75	0.08	3	11	T1B-*b* 2	75	0.08	3	12
T1B-*b* 1†	75	0.19	10	22	T1B-*b* 2†	75	0.21	11	26
T1A-*c* 1	75	0.21	12	22	T1A-*c* 2	75	0.24	13	26
T1B-*c* 1	75	0.08	4	10	T1B-*c* 2	75	0.08	3	11
T1B-*c* 1†	75	0.23	13	24	T1B-*c* 2†	75	0.23	12	27

Table 3. Overall factoid results for our sixteen Spanish runs

Task, System	#Q	MRR	TOP1	TOP5	Task, System	#Q	MRR	TOP1	TOP5
T2A-*m* 1	44	0.24	7	16	T2A-*m* 2	44	0.29	8	20
T2B-*m* 1	44	0.34	8	20	T2B-*m* 2	44	0.33	12	20
T2A-*a* 1	44	0.15	6	8	T2A-*a* 2	44	0.20	8	12
T2B-*a* 1	44	0.14	6	6	T2B-*a* 2	44	0.24	10	12
T2A-*b* 1	44	0.18	6	12	T2A-*b* 2	44	0.20	7	13
T2B-*b* 1	44	0.20	7	12	T2B-*b* 2	44	0.20	7	12
T2A-*c* 1	44	0.22	9	12	T2A-*c* 2	44	0.20	8	11
T2B-*c* 1	44	0.13	5	8	T2B-*c* 2	44	0.21	9	10

answer and nothing more, and it is supported by the corresponding document. If an answer was incomplete or it included more information than necessary or the document did not provide the justification for the answer, the answer was considered incorrect.

Tables 2 and 3 summarize our overall results for factual questions in English and Spanish. It shows MRR, TOP1 and TOP5 scores for each track and run as defined previously.

Table 4 contains a statistical error analysis of our system covering the QC, PR and AE parts. It deals only with factoid questions with non–nil answers. The meaning of each column is the following. Q: number of factual question. QC: number of questions with answer type correctly detected by QP. PR: number of question where at least on passage with the correct answer was retrieved. QC&PR: number of questions with correct answer type and correct passage retrieval. C.NE: number of questions where the retrieved passages contain the correct answer tagged as a NE of the right type (specified by the QC module), so it is a candidate answer for the AE module. TOP5 non-nil: number of question with non-nil answer correctly answered by our system in the TOP5 candidates.

Table 4. Error analysis of the QA system components

Track	Run	Q	QC	PR	QC&PR	C.NE	TOP5 non-Null	TOP1 non-Null
T1A-*m*	1	65	49	50	41	41	28	11
T1A-*a*	1	65	49	48	38	38	26	11
T1A-*b*	1	65	49	46	34	29	23	11
T1A-*c*	1	65	49	40	30	30	20	10
Avg. Loss			-23%	-28%	-44%	-3%	-29%	-55%
T1B-*m*	1	65	48	49	38	41	26	9
T1B-*a*	1	65	48	49	38	36	25	12
T1B-*b*	1	65	48	49	37	31	21	9
T1B-*c*	1	65	48	43	32	30	23	12
Avg. Loss			-26%	-27%	-44%	-0.05%	-31%	-56%
T1A-*m*	2	65	49	50	41	41	28	12
T1A-*a*	2	65	49	48	38	38	27	12
T1A-*b*	2	65	49	46	34	34	26	13
T1A-*c*	2	65	49	40	30	30	23	11
Avg. Loss			-23%	-28%	-44%	0%	-27%	-53%
T1B-*m*	2	65	48	49	38	41	25	11
T1B-*a*	2	65	48	49	38	39	27	9
T1B-*b*	2	65	48	49	37	40	25	10
T1B-*c*	2	65	48	49	37	39	26	11
Avg. Loss			-26%	-25%	-36%	0%	-35%	-61%
T2A-*m*	1	44	39	35	30	24	10	3
T2A-*b*	1	44	39	33	28	21	11	5
T2A-*a*	1	44	39	36	31	24	9	5
T2A-*c*	1	44	39	36	31	24	11	8
Avg. Loss			-11%	-20%	-31%	-22%	-56%	-48%
T2B-*m*	1	44	27	36	19	19	10	4
T2B-*a*	1	44	27	36	19	19	6	4
T2B-*b*	1	44	27	33	17	16	9	4
T2B-*c*	1	44	27	36	20	18	6	3
Avg. Loss			-38%	-19%	-46%	-4%	-57%	-51%
T2A-*m*	2	44	39	35	30	27	15	5
T2A-*a*	2	44	39	36	31	28	12	7
T2A-*b*	2	44	39	33	28	26	13	6
T2A-*c*	2	44	39	36	31	29	11	8
Avg. Loss			-11%	-20%	-31%	-8%	-53%	-49%
T2B-*m*	2	44	27	36	19	19	13	7
T2B-*a*	2	44	27	36	19	19	11	8
T2B-*b*	2	44	27	33	17	17	9	5
T2B-*c*	2	44	27	36	20	20	8	6
Avg. Loss			-38%	-19%	-46%	0%	-45%	-36%

There is an "Avg. Loss" row for each task and run that shows the performance loss (averaged in all transcripts) introduced by each module. The loss of QC, PR and QC&PR columns is relative to the total number of questions Q, for the rest of columns it is relative to the previous step. Note that this numbers have been gathered using an automatic application and some disagreement between in the selection of factoid questions may exist, therefore the TOP5 scores in this table may differ slightly from the official QAST scores.

In Table 2 we can see that moving from transcript m to a implies a loss of 10 points in TOP5 score for T1. For T2 this loss is as much as 50 points. But subsequent increases of WER in transcripts b and c have a low impact in our performance. According to the QAST 2009 overview paper [7], the incidence of WER rates in our system is less severe than in runs from other participants but our initial results in m track are also lower.

We should note that the official QAST evaluation shows an important loss of performance in T1 scenario when using the question set B (Table 2). This was caused by an error of encoding in our question files that prevented a correct question classification in most cases. This error has been fixed afterwards evaluation, the correct runs are those marked with a †. These correct runs are used for error analysis in Table 4.

Table 4 shows that sponatenous questions are not more difficult to classify than written questions for English. For the Spanish runs T2 there is a notable performance drop in QC (from 39 to 27). This must be blamed on our machine learning question classifier, wich is an straightforward adaptation of the English one. Additionally, we note that the classification of written questions is better for T2 question set than T1 question set. This suggests that in this evaluation T1 questions are more domain–specific than the others.

The difference between runs number 1 and 2 is that number 2 uses hand–tagged NEs instead of our automatic NERC. The results show that it has little impact on performance. In Table 4 we can see that most of the correct answers retrieved by our PR module are annotated with the correct entity. It is shown by the small difference between QC&PR and C.NE columns. Using hand-tagged NEs improves slightly the results for TOP5 and TOP1, probably because it filters out incorrect candidates and the AE process becomes easier. As we have seen in Table 3, the $F_{\beta=1}$ score of our NERC models is below 70% but this poor performance does not reflect in the final QA results. We think that hand-tagged NEs does not improve the results due to two facts. On one hand, the NERC we have developed is useful enough for this task even having poor $F_{\beta=1}$ scores, and on the other hand, there is a probable disagreement between the humans who tagged the NEs and the humans who wrote the questions.

One of the main issues of our QAST 2009 system is the poor performance of the AE module. More than 25% of the correctly retrieved and tagged answers aren't correctly extracted in T1, and more than 50% are lost in T2. In fact, AE is the main source of errors in T2, more than the combination of both PR and QC. This is a big difference with the results achieved in 2008 evaluation, where

AE was of high accuracy. This means that our answer selecting heuristics may be more domain–dependant than we knew and they should be fine–tuned for this task.

5 Conclusions

This paper describes UPC's participation in the CLEF 2009 Question Answering on Speech Transcripts track. We submitted runs for all English and Spanish scenarios. In this evaluation we analyzed the impact of using *gold–standard* NEs with using a far from perfect NERC.

We have developed a new NERC designed for speech transcripts that shows results competitive with *gold-standard* NEs when used in Question Answering.

The results achieved in the different scenarios and tasks are not the top ones. But there is little degradation due to ASR effects thus showing that our QA system is highly robust to transcript errors, being this one of the main focuses of the QAST evaluation.

Acknowledgements

This work has been funded by the Spanish Ministry of Science and Technology (TEXTMESS project, TIN2006–15265–C06–05).

References

1. Comas, P.R., Turmo, J.: Robust question answering for speech transcripts: UPC experience in QAst 200. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 492–499. Springer, Heidelberg (2009)
2. Comas, P.R., Turmo, J.: Spoken document retrieval based on approximated sequence alignment. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 285–292. Springer, Heidelberg (2008)
3. Freund, Y., Schapire, R.: Large margin classification using the perceptron algorithm. In: COLT 1998: Proceedings of the Eleventh Annual Conference on Computational Learning Theory (1998)
4. Li, X., Roth, D.: Learning question classifiers: The role of semantic information. Journal of Natural Language Engineering (2005)
5. Pasca, M.: High-performance, open-domain question answering from large text collections. PhD thesis, Southern Methodist University, Dallas, TX (2001)
6. Surdeanu, M., Dominguez-Sal, D., Comas, P.R.: Design and performance analysis of a factoid question answering system for spontaneous speech transcriptions. In: Proceedings of the International Conference on Spoken Language Processing, INTERSPEECH 2006 (2006)
7. Turmo, J., Comas, P.R., Rosset, S., Galibert, O., Moreau, N., Mostefa, D., Rosso, P., Buscaldi, D.: Overview of QAST 2009. In: Proceedings of the CLEF 2009 Workshop on Cross-Language Information Retrieval and Evaluation (2009)

Where in the Wikipedia Is That Answer? The XLDB at the GikiCLEF 2009 Task

Nuno Cardoso, David Batista, Francisco J. Lopez-Pellicer*, and Mário J. Silva

University of Lisbon, Faculty of Sciences, LaSIGE
*IAAA, Centro Politécnico Superior (CPS) Universidad de Zaragoza, Spain
{ncardoso,dsbatista}@xldb.di.fc.ul.pt, fjlopez@unizar.es, mjs@di.fc.ul.pt

Abstract. We developed a new semantic question analyser for a custom prototype assembled for participating in GikiCLEF 2009, which processes grounded concepts derived from terms, and uses information extracted from knowledge bases to derive answers. We also evaluated a newly developed named-entity recognition module, based in Conditional Random Fields, and a new world geo-ontology, derived from Wikipedia, which is used in the geographic reasoning process.

1 Introduction

We have been researching methods, algorithms and a software architecture for geographic information retrieval (GIR) systems since 2005 [1]. In GikiCLEF, we addressed other tasks that have not been the focus of previous GeoCLEF participations:

- Develop a new semantic question analyser module, which captures key concepts of the question's information need into workable objects, and uses those concepts to comprehend geographic restrictions and reason the correct answers;
- Develop a new world geographic ontology, Wiki WGO 2009, derived from Wikipedia pages, and organised according to an improved version of our geographic knowledge model;
- Evaluate HENDRIX [2], a new in-house named entity recognition software based on machine-learning approaches, that can recognise geographic locations and map them into the Wiki WGO 2009 ontology.

2 GikiCLEF Approach

We assembled a custom prototype QA system for participating in GikiCLEF, GreP (Grease Prototype), depicted in Figure 1. GreP is composed of a question analyser module and a group of knowledge resources. The GikiCLEF topics are initially parsed by the Portuguese PoS tagger PALAVRAS [3], before being fed into the question analyser, which explores multiple resources.

As GikiCLEF topics are in the form of a question, the initial task performed by the *question interpreter* (QI) is to convert such questions into object representations (*question objects*). From there, the *question reasoner* (QR) picks the best strategy to

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 305–309, 2010.

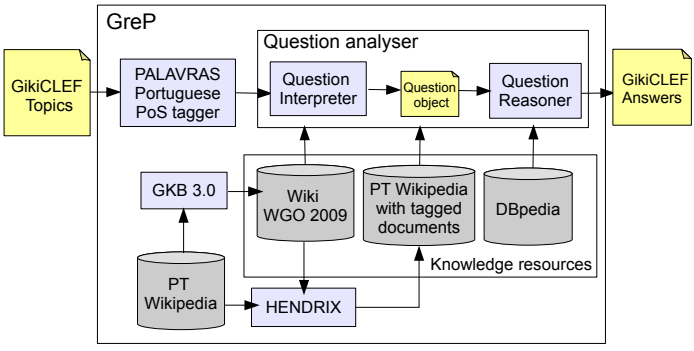


Fig. 1. Overview of GreP, the system assembled for participating in GikiCLEF

obtain the correct answers, which may trigger information extraction procedures over the raw knowledge resources. The QR output is the answers and their justifications, which are converted into the GikiCLEF run format. A distinct characteristic of this approach is that it works with concepts, not terms, in the key steps. GreP includes three knowledge resources: i) the Wiki WGO 2009 geographic ontology, derived from the Portuguese Wikipedia, ii) the DBpedia v3.2 dataset [4], derived from the English Wikipedia, and iii) the Portuguese Wikipedia, with tagged documents by HENDRIX.

2.1 Question Interpreter

The question interpreter (QI) converts a natural language question into *question objects*, a machine-interpretable object representing that question. A question object is composed of the following elements:

Subject, the class of expected answers. A subject can be grounded as i) a DBpedia resource that represents a Wikipedia category, ii) a DBpedia ontology class, or iii) a semantic classification from HAREM categorization [5], on this preferential order.
Conditions, a criteria list that filters candidate answers. Each condition is composed of i) a DBpedia ontology property, ii) an operator and iii) a DBpedia resource.
Expected Answer Type (EAT), defines properties that the answers must have.

The question object is generated by applying a set of pattern rules over the PoS tags and the question terms. Given for instance, the question "Which Romanian writers were born in Bucharest?", the QI would perform as follows:

Ground the subject and EAT: a first set of pattern rules detects "Romanian writers" as a subject, grounded to the DBpedia resource `http://dbpedia.org/resource/Category:Romanian_writers`, which is derived from the corresponding Wikipedia's category page. Afterwards, another set of pattern rules maps the subject to the EAT, that is, the answers must have the "`Category:Romanian_writers`". If the subject cannot be mapped to a DBpedia resource, it is mapped to the closest DBpedia ontology class (for example, "`Writers`") or to a HAREM category (for example, `PERSON/INDIVIDUAL`).

Ground conditions: A set of patterns ground the condition in the example, "were born in Bucarest", into a property `http://dbpedia.org/ontology/birthplace` and a referent entity to `http://dbpedia.org/resource/Bucharest`.

2.2 Question Reasoner

The question reasoner (QR) processes the grounded concepts within the question object, aiming to resolve it into a list of answers. Depending on the elements present in the question object, the QR task decides the best strategy to generate and validate those answers, which consists of a pipeline of SPARQL queries made to the knowledge resources. In the given example, for a question object with an EAT given by `Category:Romanian_writers`, a single condition described by a property `dbpedia-owl:birthPlace` and a referent geographic entity `http://dbpedia.org/resource/Bucharest`, the QR strategy solves the question by issuing the following SPARQL query to the DBpedia dataset:

```
SELECT ?RomanianWriters WHERE {?RomanianWriters
 skos:subject <http://dbpedia.org/resource/Category:Romanian_writers> .
 ?RomanianWriters dbpedia-owl:birthplace <http://dbpedia.org/resource/Bucharest>}
```

2.3 Wiki WGO 2009

Wiki WGO 2009 [6] is a geospatial ontology derived from a SQL snapshot of the Portuguese Wikipedia. It was generated with our in-house system, which integrates geographic and Web-related data collected from heterogeneous sources [1]. Wiki WGO 2009 contains 136,347 geographic features with associated named (484,097 names in 8 languages). It also contains 2,444 feature types, many of them in the "something in some place" format with transitive *part-of* relationships.

The QR module computes a set of resources spatially related to the referent entity from Wiki WGO 2009. In the above example, if the condition was "born in Romania", the QR would rewrite the condition as "*born in ?X*" and "*?X in Romania*". The last condition is resolved against Wiki WGO 2009, and the answers are applied to resolve the first condition.

2.4 HENDRIX

HENDRIX is a named entity recognition system developed in-house [2]. It uses Minorthird's Conditional Random Fields (CRF) implementation, a supervised machine learning technique for text tagging and information extraction [7]. HENDRIX was trained to recognise places, organisations, events and people. It uses the Wiki WGO 2009 ontology to detect relations between named entities (NEs) tagged as geographic locations.

The CRF model was trained with manually-tagged collections of Portuguese documents used in the HAREM joint NER evaluation for Portuguese [5]. Of three document collections available, two were used for the training phase and for evaluation of HENDRIX's performance. We used in HENDRIX a single CRF model for extracting NEs of the four different entity types (PERSON, PLACE, EVENT, ORGANIZATION). The trained model was then used to extract NEs from Portuguese Wikipedia articles. All the

extracted entities tagged as PLACEs were afterwards used for identification of semantic relationships between the entities, using the Wiki WGO 2009 ontology.

We had to resort to using an on-demand tagging strategy, as it was not possible to tag the whole Portuguese Wikipedia in time with the available computational resources. This limitation prevented the use of more complex information extraction approaches that we had envisioned. We observed that there is a significant fraction of NEs that are correctly extracted but incorrectly categorised. This prompted us to consider using a single CRF model trained separately for each entity type, instead of one that labels NEs of four different types.

3 Lessons Learned

The single run submitted to GikiCLEF had 1161 answers, and 332 of them were assessed as correct and justified. Within the scope of a *post hoc* analysis made over the GikiCLEF answers and the Wikipedia collections [8], these answers correspond to 60 entities, out of an universe of 262 entities found by all GikiCLEF systems.

The results were below our expectations, and can be primarily explained by: i) a limited coverage of solutions by the Portuguese Wikipedia, which contains Wikipedia pages for roughly half of the solutions, making maximum recall limited at start, and ii) a significant amount of justifications that can be found in the article's body only, and could not be accessed by SPARQL queries; the DBpedia dataset is created from Wikipedia infoboxes, which contain relevant information only to a limited number of topics. Nevertheless, we're giving our first steps towards a GIR approach that relies on a reasoning core over grounded concepts, not terms, which uses raw knowledge resources and geographic ontologies to resolve user information needs before the retrieval phase.

Acknowledgements

This work was supported by FCT for its LASIGE Multi-annual support, GREASE-II project (PTDC/EIA/73614/2006) and Nuno Cardoso's scholarship (SFRH/BD/45480/2008). Francisco J. Lopez-Pellicer has been supported by the Spanish Government (TIN2007-65341 and PET2008_0026) and the Aragon Government (PI075/08).

References

1. Silva, M.J., Martins, B., Chaves, M., Afonso, A.P., Cardoso, N.: Adding Geographic Scopes to Web Resources. CEUS - Computers Enviroment and Urban Systems 30, 378–399 (2006)
2. Batista, D.: Prospecção de conceitos geográficos na web. Master's thesis, University of Lisbon, Faculty of Sciences (2009) (in Portuguese)
3. Bick, E.: The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. PhD thesis, University of Aarhus, Aarhus, Denmark (2000)
4. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
5. Mota, C., Santos, D.: Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM. Linguateca (2009)

6. Lopez-Pellicer, F.J., Chaves, M., Rodrigues, C., Silva, M.J.: Geographic Ontologies Production in Grease-II. Technical Report DI/FCUL TR 09-18, Faculty of Sciences, University of Lisbon (2009), doi:10455/3256
7. Cohen, W.W.: MinorThird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data (2004), http://minorthird.sourceforge.net
8. Cardoso, N.: GikiCLEF topis and Wikipedia articles: did they blend? In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 318–321. Springer, Heidelberg (2010)

Recursive Question Decomposition for Answering Complex Geographic Questions

Sven Hartrumpf[1] and Johannes Leveling[2]

[1] Intelligent Information and Communication Systems (IICS),
University of Hagen (FernUniversität in Hagen),
Hagen, Germany
sven.hartrumpf@fernuni-hagen.de
[2] Centre for Next Generation Localisation (CNGL)
School of Computing
Dublin City University
Dublin 9, Ireland
jleveling@computing.dcu.ie

Abstract. This paper describes the GIRSA-WP system and the experiments performed for GikiCLEF 2009, the geographic information retrieval task in the question answering track at CLEF 2009. Three runs were submitted. The first one contained only results from the InSicht QA system; it showed high precision, but low recall. The combination with results from the GIR system GIRSA increased recall considerably, but reduced precision. The second run used a standard IR query, while the third run combined such queries with a Boolean query with selected keywords. The evaluation showed that the third run achieved significantly higher mean average precision (MAP) than the second run. In both cases, integrating GIR methods and QA methods was successful in combining their strengths (high precision of deep QA, high recall of GIR), resulting in the third-best performance of automatic runs in GikiCLEF. The overall performance still leaves room for improvements. For example, the multilingual approach is too simple. All processing is done in only one Wikipedia (the German one); results for the nine other languages are collected by following the translation links in Wikipedia.

1 Introduction

GIRSA-WP (GIRSA for Wikipedia) is a fully-automatic, hybrid system combining methods from question answering (QA) and geographic information retrieval (GIR). It merges results from InSicht, an open-domain QA system [1], and GIRSA, a system for textual GIR [2]. GIRSA-WP has already participated in the preceding pilot task, GikiP 2008 [3,4], and was improved based on this and other evaluations.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 310–317, 2010.

2 System Description

2.1 GIRSA-WP Subsystems

The GIRSA-WP system used for GikiCLEF 2009 integrates two basic systems: a deep (text-semantic) QA system (InSicht) and a GIR system (GIRSA, GIR with semantic annotation). Each question is processed by both basic systems; GIRSA-WP filters their results semantically to improve precision and combines both result streams yielding a final result of Wikipedia article names, additional supporting article names (if needed), and supporting text snippets.

The semantic filter checks whether the expected answer type (EAT) of the question and the title of a Wikipedia article are semantically compatible. This technique is widely known from QA for typical answer types such as PERSON, ORGANIZATION, or LOCATION. In our system, a concept (a disambiguated word) corresponding to the EAT is extracted from the question. The title of each candidate article is parsed by a syntactico-semantic parser for German [5]. The resulting semantic representations (comprising the sort and the semantic features, see [6] for details on the semantic representation formalism MultiNet) of the representations from the question and from the article title are unified. If this unification succeeds, the candidate article is kept; otherwise it is discarded. For example, from topic GC-2009-06 (*Which Dutch violinists held the post of concertmaster at the Royal Concertgebouw Orchestra in the twentieth century?*), the concept extracted as EAT is *violinist.1.1*, whose semantic representation belongs to the class human (*human-object* in MultiNet). There are 87 such semantic classes, which can also be combined to form disjunctive expressions for underspecification or for so-called semantic molecules (or semantic families).

The retrieval in the GIR system works on the first few (two or three) sentences of the Wikipedia articles. Geographic names and location indicators (e.g. name variants and adjectives corresponding to toponyms) in the articles were automatically annotated and normalized (see [2] for a discussion of this approach). As a result of our participation in GikiCLEF last year, we found that the full Wikipedia articles may be too long and indexing on a per-sentence basis does not provide enough context for matching. Therefore, we focused on the most important parts of the Wikipedia articles (to increase precision for GIRSA), and changed to full-document indexing.

For the GikiCLEF 2009 experiments, the questions were analyzed by InSicht's parser and sent to GIRSA and InSicht. In GIRSA, the top 1000 results were retrieved, with scores normalized to the interval $[0, 1]$. On average, GIRSA returned 153 and 395 documents per question for run 2 and run 3, respectively (see Sect. 3). For results returned by both GIRSA and InSicht, the maximum score was chosen (combMAX, [7]). Results whose score was below a given threshold were discarded and the semantic filter was applied to the remaining results.

2.2 External Knowledge

To obtain multilingual results, the German article names were 'translated' to the nine other languages using the Wikipedia linking between languages. Besides the

inter-wiki links, GIRSA-WP uses one further information type from Wikipedia: the categories assigned to articles. Note that other Wikipedia information types like intra-wiki (i.e. inter-article) links and Internet links are still ignored.

For the first time, two resources that contain structured information and are derived directly (categories) or indirectly (DBpedia) from Wikipedia were integrated into GIRSA-WP. The direct source of categories assigned to articles was exploited by extracting categories from the Wikipedia XML file. The resulting relations of the form *in_category*(⟨*article_title*⟩, ⟨*category*⟩) were reformulated in the following form: ⟨*article_title*⟩ *ist ein/ist eine/ . . .* ⟨*category*⟩*/'*⟨*article_title*⟩ *is a . . .* ⟨*category*⟩*'*. Some automatic corrections for frequent cases where the text would be syntactically and/or semantically incorrect were implemented. The remaining errors were largely unproblematic because the processing by InSicht's parser detects them and avoids incorrect semantic networks. In this way, 1.1 million semantic networks were generated for 1.5 million sentences derived from around 2 million *in_category* relations.

The DBpedia data is integrated in a similar way into GIRSA-WP by rephrasing it in natural language. Specifically, version 3.2 of the file infobox_de.nt, the infobox information from the German Wikipedia encoded in N-Triples, a serialization of RDF was processed (see http://wiki.dbpedia.org/ for details). As there are many different relations in DBpedia, only some frequent and relevant relations are covered currently. Each selected relation (currently 19) is associated with an abstract relation (currently 16) and a natural language pattern. For example, the triple

```
<http://dbpedia.org/resource/Andrea_Palladio>
<http://dbpedia.org/property/geburtsdatum>
"1508-11-08"^^<http://www.w3.org/2001/XMLSchema#date>
```

is translated to *Andrea Palladio wurde geboren am 08.11.1508./'Andrea Palladio was born on 08.11.1508.'* This generation process led to around 460,000 sentences derived from around 4,400,000 triples in the DBpedia file.

The detour of translating structured information resources to natural language is performed with the goal to treat all resources in the same way, i.e. parsing them to obtain their representation as semantic networks. Hence, the results can be used in the same way, e.g. for reasoning and to provide answer support. In addition, the parser is able to resolve ambiguities; for example, names referring to different kinds of entities that had to be disambiguated explicitly on the structured level otherwise.

The QA system (InSicht) compares the semantic representation of the question and the semantic representations of document sentences. To go beyond exact matching, InSicht applies many techniques, e.g. coreference resolution, query expansion by inference rules and lexico-semantic relations, and splitting the query semantic network at certain semantic relations. In the context of GikiCLEF, InSicht results (which are generated answers in natural language) must be mapped to Wikipedia article names; if this is not straightforward, the article name of the most important support is taken.

2.3 Recursive Question Decomposition

InSicht employed a new special technique called *question decomposition* (or *query decomposition*, see [8] for details) for GeoCLEF 2007, GeoCLEF 2008, and GikiP 2008. An error analysis showed that sometimes it is not enough to decompose a question once. For example, question GC-2009-07 (*What capitals of Dutch provinces received their town privileges before the fourteenth century?*) is decomposed into the subquestion *Name capitals of Dutch provinces.* and revised question *Did* $\langle$*SubA(nswer)*$^1\rangle$ *receive its town privileges before the fourteenth century?* Unfortunately, the subquestion is still too complex and unlikely to deliver many (if any) answers. This situation changes if one decomposes the subquestion further into a subquestion (second level) *Name Dutch provinces.* and a revised question (second level) *Name capitals of* $\langle$*SubA(nswer)*$^2\rangle$. InSicht's processing of question GC-2009-07 is illustrated in more detail in Fig. 1. For brevity and better readability, additional question reformulation phases and intermediate stages have been omitted and the supporting texts are shortened and not translated. All subquestions and revised questions are shown in natural language, while the system operates mostly on the semantic (network) level.

Question decomposition, especially in its recursive form, is a very powerful technique that can provide answers and justifications for complex questions. However, the success rates at each decomposition combine in a multiplicative way. For example, if the QA system has an average success rate of 0.5, a double decomposition as described above (leading to questions on three levels) will have an average success rate of 0.125 ($= 0.5 \cdot 0.5 \cdot 0.5$).

3 Experiments

We produced three runs with the following experiment settings:

- Run 1: only results from InSicht.
- Run 2: results from InSicht and GIRSA, using a standard query formulation and a standard IR model (tf-idf) in GIRSA.
- Run 3: results from InSicht and GIRSA, using a Boolean conjunction of the standard query formulation employed for GIRSA and (at most two) keywords extracted from the topic.

4 Evaluation and Discussion

InSicht achieved a higher precision than GIRSA: 0.7895 compared to 0.1076 and 0.1442 for run 2 and run 3, respectively (see Table 2). The definition of the GikiCLEF score and other task details can be found in [9]. But InSicht's low recall (only 30 correct answers compared to 107 and 142 correct answers for run 2 and run 3, respectively) is still problematic as has already been seen in similar evaluations, e.g. GikiP 2008. As intended, InSicht aims for precision, GIRSA for recall, and GIRSA-WP tries to combine both in an advantageous way.

Table 1. Illustration of successful recursive question decomposition for topic GC-2009-07. The superscript designates the level of recursion, the subscript distinguishes alternatives on the same level of recursion.

Q^0	*Welchen Hauptstädten niederländischer Provinzen wurde vor dem vierzehnten Jahrhundert das Stadtrecht gewährt?* *'What capitals of Dutch provinces received their town privileges before the fourteenth century?'*
$SubQ^1 \leftarrow Q^0$	*Nenne Hauptstädte niederländischer Provinzen.* *'Name capitals of Dutch provinces.'*
$SubQ^2 \leftarrow SubQ^1$	*Nenne niederländische Provinzen.* *'Name Dutch provinces.'*
$SubA_1^2 \leftarrow SubQ^2$	*Zeeland* (support from article *1530*: *Besonders betroffen ist die an der Scheldemündung liegende niederländische Provinz Zeeland.*)
$SubA_2^2 \leftarrow SubQ^2$	*Overijssel* ...
⋮	
$RevQ_1^1 \leftarrow SubA_1^2 + SubQ^1$	*Nenne Hauptstädte von Zeeland.* *'Name capitals of Zeeland.'*
$RevA_1^1 \leftarrow RevQ_1^1$	*Middelburg* (support from article *Miniatuur Walcheren*: *... in Middelburg, der Hauptstadt von Seeland (Niederlande).*; note that the orthographic variants *Zeeland*/*Seeland* are identified correctly)
$SubA_1^1 \leftarrow RevA_1^1$	*Middelburg* (note: answer to revised question can be taken without change)
$RevQ_1^0 \leftarrow Q^0 + SubA_1^1$	*Wurde Middelburg vor dem vierzehnten Jahrhundert das Stadtrecht gewährt?* *'Did Middelburg receive its town privileges before the fourteenth century?'*
$RevA_1^0 \leftarrow RevQ_1^0$	*Ja.*/*'Yes.'* (support from article *Middelburg*: *1217 wurden Middelburg durch Graf Willem I. ... die Stadtrechte verliehen.*)
$A_1^0 \leftarrow RevA_1^0 + SubA_1^1$	*Middelburg* (support: three sentences, here from different articles, see supports listed in previous steps)

In order to investigate the complementarity of GIRSA and InSicht, two experimental runs were performed after the campaign. In run 4 and 5, only results from GIRSA are included; the settings correspond to the ones from run 2 and run 3, respectively. The number of correct answers (compare run 2 and sum of run 1 and 4; compare run 3 and sum of run 1 and 5) shows that the overlap of GIRSA and InSicht is minimal: only 1 correct answer is shared. Hence, the combination of both systems is very effective. The results indicate also that the combination of the two systems profits from keeping most of InSicht's correct results and discarding some incorrect results from GIRSA.

Table 2. Evaluation results for the three official GIRSA-WP runs and two experimental runs

Run	System	Answers	Correct answers	Precision	GikiCLEF score
1	InSicht	38	30	**0.7895**	**24.7583**
2	InSicht+GIRSA	994	107	0.1076	14.5190
3	InSicht+GIRSA	985	**142**	0.1442	23.3919
4	GIRSA	964	78	0.0809	7.8259
5	GIRSA	961	113	0.1176	15.0473

We made the following general observations:

Complexity of Questions. GikiCLEF topics are open-list questions and do not include factoid or definition questions. On average, GikiCLEF questions seem to be harder than QA@CLEF questions from the years 2003 till 2008. Especially the presence of temporal and spatial (geographical) constraints in GikiCLEF questions poses challenges for QA and GIR techniques, which cannot be met successfully by shallow (i.e. syntactically-oriented) natural language processing or traditional IR techniques alone.

Combination of standard and Boolean IR. As the GikiCLEF topics resemble open list questions, the aim of the GIR approach was to retrieve results with a high initial precision. The use of the query formulation which combines keywords extracted from the query with a standard IR query (run 3) increases precision (+34%) and recall (+33%) compared to the standard IR query formulation (run 2).

Question decomposition. As our question decomposition experiments indicate, correct answers can often not be found in one step; instead, subproblems must be solved or subquestions must be answered in the right order. For some topics, a promising subquestion leads to many answers (for example, the subquestion *Nenne Städte in Deutschland./'Name cities in Germany.'* for topic GC-2009-47), which cannot be efficiently handled for the revised questions so that correct answers are missed.

Abstract indexing. Indexing shorter (abstracted) Wikipedia articles returned a higher number of correct results (which was tested on some manually annotated data before submission). Similarly, the annotation of geographic entities in the documents (i.e. conflating different name forms etc.) ensured a relatively high recall.

Multilingual Results. The system's multilingual approach is too simple because it relies only on the Wikipedia in one language (German) and adds results by following title translation links to other languages. For eleven GikiCLEF topics (5, 10, 15, 16, 18, 24, 26, 27, 28, 36, and 39) no articles in German were assessed as relevant. Therefore for questions that have no or few articles in German, relevant articles in other languages cannot be found. Processing the Wikipedia articles in parallel for another language in the same way also will allow to find subanswers supported by articles in other languages, i.e. the supporting texts may not only be distributed among different articles of only one languages, but also among articles in different languages.

5 Future Work

Some resources are not yet exploited to their full potential. For example, almost half of the category assignments are ignored (see Sect. 2). Similarly, many attribute-value pairs from infoboxes in DBpedia are not covered by GIRSA-WP currently. The cross-language aspect should be improved by processing at least one more Wikipedia version, preferably the largest one: the English Wikipedia.

Acknowledgments

This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/CE/I1142.

References

1. Hartrumpf, S.: Question answering using sentence parsing and semantic network matching. In: Peters, C., Clough, P., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B. (eds.) CLEF 2004. LNCS, vol. 3491, pp. 512–521. Springer, Heidelberg (2005)
2. Leveling, J., Hartrumpf, S.: Inferring location names for geographic information retrieval. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 773–780. Springer, Heidelberg (2008)
3. Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J., Skalban, Y.: Getting geographical answers from Wikipedia: the GikiP pilot at CLEF. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, Springer, Heidelberg (2009)

4. Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J., Skalban, Y.: GikiP at GeoCLEF 2008: Joining GIR and QA forces for querying Wikipedia. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 894–905. Springer, Heidelberg (2009)
5. Hartrumpf, S.: Hybrid Disambiguation in Natural Language Analysis. Der Andere Verlag, Osnabrück (2003)
6. Helbig, H.: Knowledge Representation and the Semantics of Natural Language. Springer, Berlin (2006)
7. Fox, E.A., Shaw, J.A.: Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2). NIST Special Publication 500-215, pp. 243–252. National Institute for Standards and Technology (1994)
8. Hartrumpf, S.: Semantic decomposition for question answering. In: Ghallab, M., Spyropoulos, C.D., Fakotakis, N., Avouris, N. (eds.) Proceedings of the 18th European Conference on Artificial Intelligence (ECAI), Patras, Greece, pp. 313–317 (July 2008)
9. Santos, D., Cabral, L.M.: GikiCLEF: Expectations and lessons learned. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 212–222. Springer, Heidelberg (2010)

GikiCLEF Topics and Wikipedia Articles: Did They Blend?

Nuno Cardoso

University of Lisbon, Faculty of Sciences, LaSIGE
and Linguateca, Oslo node, SINTEF ICT, Norway
ncardoso@xldb.di.fc.ul.pt

Abstract. This paper presents a post-hoc analysis on how the Wikipedia collections fared in providing answers and justifications to GikiCLEF topics. Based on all solutions found by all GikiCLEF participant systems, this paper measures how self-sufficient the particular Wikipedia collections were to provide answers and justifications for the topics, in order to better understand the recall limit that a GikiCLEF system specialised in one single language has.

1 Introduction

The GikiCLEF 2009 evaluation track [1] proposed a multilingual answer task using Wikipedia snapshots as a collection. Although the author participated in GikiCLEF as a co-organiser, gathering the Wikipedia snapshots and SQL dumps for 10 languages from June 2008, and generating the XML version of the GikiCLEF collections, he had obviously no say in the topic choice. As a GikiCLEF participant within the XLDB team [2], we used the Portuguese topics and searched for answers in the Portuguese Wikipedia, using language links afterwards to collect answers for other languages. The GikiCLEF results show that 332 correct answers were found out of 1161, which was below our expectations.

In overall, the GikiCLEF systems submitted a total of 18152 unique answers, from which 1008 of them were considered correct and justified by the GikiCLEF assessors. This answer pool accounts for 262 *solutions*, that is, unique entities. For example, "Italy" is one solution for a given topic, regardless of the number of answers corresponding to Italy from different languages. XLDB managed to find only 60 solutions out of 262.

With only the GikiCLEF assessment results, it's not possible to determine which share of those 202 unseen solutions can be due to XLDB's approach or malfunction, and which share was simply not available in the Portuguese collection. In order to find this share, a post-hoc analysis over all 262 solutions was made, by checking the presence of such answers and justification excerpts in all Wikipedia languages on the Wikipedia site in June and July 2009 (roughly one year from the GikiCLEF collection date). This analyses was performed by the author who is a Portuguese native speaker, fluent in English and fairly familiar with French and Spanish, using online translation tools to search for justifications in other languages (which will likely add some questionable judgements on the analysis), so any conclusions drawn from this post-hoc analysis must take this fact in consideration.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 318–321, 2010.

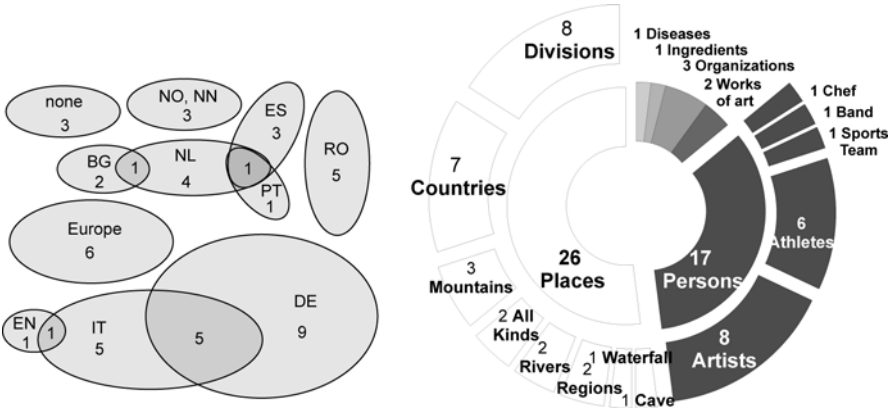


Fig. 1. In the left, the 50 GikiCLEF topics organised by language-biased balloons, and in the right, the expected answer type distribution of the topics

Nonetheless, this analysis shed some light on how well the particular Wikipedia collection "blended in" with the GikiCLEF topics, namely:

- How self-sufficient is a language in providing answers and justifications alone for the topics, and how are the answers distributed over the languages in Wikipedia?
- Does it pay to go, for example, to the Romanian Wikipedia search for answers about Romanian writers, or can any given language be chosen for that (for example, Portuguese), or does it pay to always shift to the English Wikipedia?
- If a system is only capable of extracting information in a given set of languages, what's the maximum answer recall for that scenario?

2 Answer Distribution over Wikipedia

Figure 1 illustrates the GikiCLEF topic distribution regarding language-bias and expected answer types, where we can observe that they are in overall strongly-biased for non-English languages, and dominated by open list questions that expect places and persons. Figure 2 shows the number of solutions found per topic, with an average of 5 solutions per topic. There are 145 answers that are places (55.3%), which is a common answer type for topics with a high number of solutions.

Figure 3 divides the 262 solutions, for each language, in three parts: i) *justified answer*, where the solution has a Wikipedia page and the justification is present anywhere on that language's Wikipedia snapshot, ii) *just the answer*, where the solution has a Wikipedia page, but that language's Wikipedia snapshot does not have any explicit evidence that indicates that the answer is correct (according to my judgement), and iii) *No Answer*, where that language's Wikipedia does not even have a page for that solution. From it we can read that, for instance, a system that uses only the Portuguese Wikipedia (like XLDB's system) can find a maximum of 1/4 of the total number of answers, or

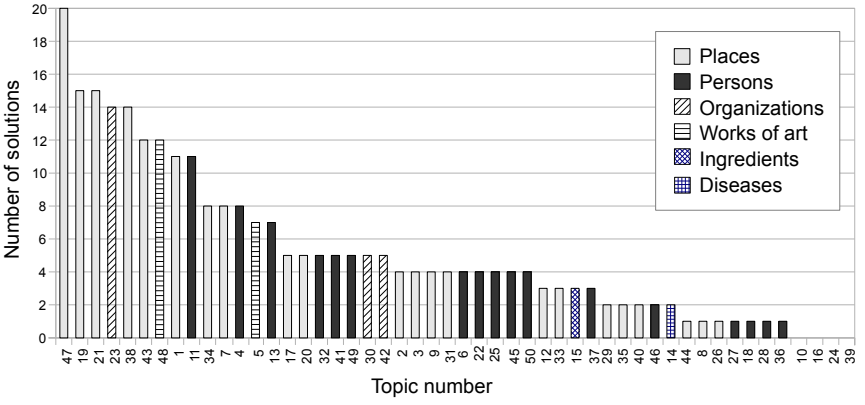


Fig. 2. Number of solutions per topic, colored by expected answer type

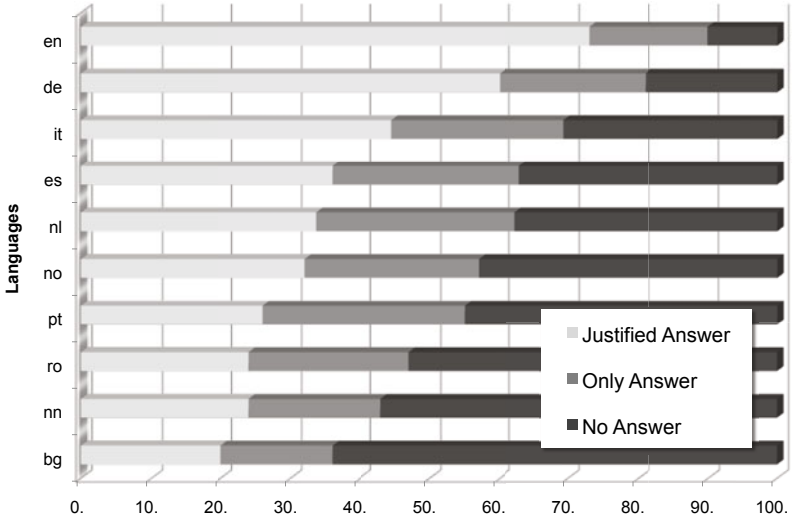


Fig. 3. Language self-sufficiency on answers and justifications

1/2 if it manages to point to other language's Wikipedia documents that are justified in themselves.

Table 1 counts the number of languages able to provide answers for each topic. Note that, apart from the 5 topics without solution found and the single topic (#14, naming diseases with dedicated European research centers, with correct answers only from the English Wikipedia), there is a large number of topics (9) that have two languages, typically the "own" language, and English. Conversely, from the 9 topics which have at least one solution in all languages, 7 of them are open list questions that expect an

Table 1. Number of languages with correct and justified answers per topic

Number of languages	0	1	2	3	4	5	6	7	8	9	10
Number of topics	5	1	9	5	4	2	2	5	4	4	9

answer of places like cities of countries, and these are well covered over the several languages in Wikipedia.

3 Concluding Remarks

As evaluation tracks such as GikiCLEF aim to measure a system's performance for a common task, it's important to determine what is the confidence level one can have that the evaluation results are a consequence of the system's capabilities, and what is the noise level generated by the evaluation task itself.

This work analyses the Wikipedia coverage in answers for GikiCLEF topics, thus giving participants an overview of the limits of the collection, and allowing them to better understand the system's performance measures in the GikiCLEF 2009 evaluation task. We observed that the solution coverage on Wikipedia languages is somehow disappointing: from all Wikipedia languages, only the English and German snapshots have over 50% of justified answers, hence a GikiCLEF system that bases its knowledge extraction process on non-English, non-German Wikipedia will have its maximum recall considerably limited.

Acknowledgements

This work is supported by FCT for its LASIGE Multi-annual support, GREASE-II project (grant PTDC/EIA/73614/2006) and a PhD scholarship grant SFRH/BD/45480/2008, and by the Portuguese Government, the European Union (FEDER and FSE) through the Linguateca project, under contract ref. POSC/339/1.3/C/NAC, UMIC and FCCN.

References

1. Santos, D., Cabral, L.M.: GikiCLEF: Expectations and lessons learned. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, Springer, Heidelberg (2010)
2. Cardoso, N., Baptista, D., Lopez-Pellicer, F.J., Silva, M.J.: Where in the Wikipedia is that answer? the XLDB at the GikiCLEF 2009 task. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 305–309. Springer, Heidelberg (2010)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)

TALP at GikiCLEF 2009

Daniel Ferrés and Horacio Rodríguez

TALP Research Center, Software Department
Universitat Politècnica de Catalunya
Jordi Girona 1-3, 08043 Barcelona, Spain
{dferres,horacio}@lsi.upc.edu

Abstract. This paper describes our experiments in Geographical Information Retrieval with the Wikipedia collection in the context of our participation in the GikiCLEF 2009 Multilingual task in English and Spanish. Our system, called gikiTALP, follows a simple approach that uses standard Information Retrieval with the Sphinx full-text search engine and some Natural Language Processing techniques without Geographical Knowdledge.

1 Introduction

In this paper we present the overall architecture of our gikiTALP IR system and we describe its main components. We also present the experiments, results and initial conclusions in the context of the GikiCLEF 2009 Monolingual English and Spanish task.

GikiCLEF 2009 is an evaluation task under the scope of CLEF. Its aim is to evaluate systems which find Wikipedia entries/documents that answer a particular information need, which requires geographical reasoning of some sort. GikiCLEF is the successor of the GikiP 2008 [1] pilot task which ran in 2008 under GeoCLEF.

For GikiCLEF, systems will need to answer or address geographically challenging topics, on the Wikipedia collections, returning Wikipedia document titles as list of answers in all languages it can find answers.

The Wikipedia collections for all GikiCLEF languages are available in three formats, HTML dump, SQL dump, and XML version. We used the SQL dump version of the English and Spanish collections (see details about these collections summarized in Table 1).

2 System Description

The system architecture has three phases that are performed sequentially: Collection Indexing, Topic Analysis, and Information Retrieval. The textual Collection Indexing has been applied over the textual collections with MySQL and the open-source full-text engine Sphinx using the Wikipedia SQL dumps.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 322–325, 2010.

Table 1. Description of the collections we used at GikiCLEF 2009

Language	#Total	#Pages	#Templates	#Categories	#Images
en	6,587,912	5,255,077	154,788	365,210	812,837
es	714,294	641,852	11,885	60,556	1

Sphinx[1] is a full-text search engine that provides fast, size-efficient and relevant full-text search functions to other applications. The indexes created with Sphinx do not have any language processing. Sphinx has two types of weighting functions: Phrase rank and Statistical rank. Phrase rank is based on a length of longest common subsequence (LCS) of search words between document body and query phrase. Statistical rank is based on classic BM25 function which only takes word frequencies into account. We used two types of search modes in Sphinx (see[2] for more information about the search mode and weighting schemes used):

- MATCH ALL: the final weight is a sum of weighted phrase ranks.
- MATCH EXTENDED: the final weight is a sum of weighted phrase ranks and BM25 weight, multiplied by 1000 and rounded to integer.

The Topic Analysis phase extracts some relevant keywords (with its analysis) from the topics. These keywords are then used by the Document Retrieval phases.

This process extracts lexico-semantic information using the following set of Natural Language Processing tools: *TnT* (POS tagger) and [2] *WordNet lemmatizer* (version 2.0) for English, and *Freeling* [3] for Spanish. These NLP tools were used by the authors in another system for Geographical Information Retrieval in GeoCLEF 2007 [4]. The language processing with these NLP tools is applied only in the queries. The Wikipedia collection is indexed without applying the stemming and stopword filtering options of Sphinx.

The retrieval is done with Sphinx and then the final results are filtered. The Wikipedia entries without Categories are discarded.

3 Experiments

For the GikiCLEF 2009 evaluation we designed a set of three experiments that consist in applying different baseline configurations (see Table 2) to retrieve Wikipedia entries (answers) of 50 geographically challenging topics.

The three baseline runs were designed changing two parameters of the system: the IR Sphinx search mode and the Natural Language Processing techniques applied over the query. The first run (gikiTALP1) do not uses any NLP processing technique over the query and the Sphinx match mode used is MATCH_ALL. The second run (gikiTALP2) uses stopwords filtering and the lemmas of the remaining words as a query and the Sphinx match mode used is MATCH_ALL. The third run (gikiTALP3) uses stopwords filtering and the lemmas of the remaining words as a query and the Sphinx match mode used is MATCH_EXTENDED.

[1] http://www.sphinxsearch.com/

[2] http://www.sphinxsearch.com/docs/current.html. Sphinx 0.9.9 documentation.

Table 2. Description of the experiments at GikiCLEF 2009

Automatic Runs	NLP in Query	Sphinx Match
gikiTALP1	-	MATCH_ALL (phrase rank)
gikiTALP2	lemma + stopwords filtering	MATCH_ALL (phrase rank)
gikiTALP3	lemma + stopwords filtering	MATCH_EXTENDED (BM25)

4 Results

The results of the gikiTALP system at the GikiCLEF 2009 Monolingual English and Spanish task are summarized in Table 3. This table has the following IR measures for each run: number of correct answers (*#Correct Answers*), *Precision*, and *Score*.

The run gikiTALP1 obtained the following scores for English, Spanish and Global: 0.6684, 0.0280, and 0.6964. Due to an unexpected error we did not produced answers for the Spanish topics in run 2 (gikiTALP2), then the results for English and global were 1,3559. The results of the scores of the run gikiTALP3 for English, Spanish and Global were 1.635, 0.2667, and 1.9018 respectively.

Table 3. TALP GikiTALP Results

run	**Measures**	**English (EN)**	**Spanish (ES)**	**Total**
run 1	#Answers	383	143	526
	#Correct answers	16	2	18
	Precision	0.0418	0.0140	0.0342
	Score	0.6684	0.0280	0.6964
run 2	#Answers	295	–	295
	#Correct answers	20	–	20
	Precision	0.0678	–	0.0678
	Score	1.3559	–	1.3559
run 3	#Answers	296	60	356
	#Correct answers	22	4	26
	Precision	0.0743	0.0667	0.0730
	Score	1.6351	0.2667	1.9018

5 Conclusions

This is our first approach to deal with a Geographical Information Retrieval task over the Wikipedia. We have used the Sphinx full-text search engine with limited Natural Language Processing processing and without using Geographical Knowledge. We obtained the best results when we have used all the NLP techniques (lemmas in the queries and stopwords filtered) and the Sphinx mode MATCH_EXTENDED without Geographical Knowledge as baseline algorithms.

In comparison with other approaches at GikiCLEF this approach was not so good. We experienced the difficulty of having good results in the task with only standard Information Retrieval and basic NLP techniques. We expect that applying Geographical Reasoning with Geographical Knowdledge Bases and using relevant information extracted from Wikipedia and its links we can boost the performance of the system.

As a future work we plan to improve the system with the following actions: 1) detect the Expected Answer Type of the topics and use the Wordnet synsets to match it with the page Categories, 2) use Geographical Knowledge in the Topic Analysis, 3) increase the use of the Wikipedia links.

Acknowledgments

This work has been supported by the Spanish Research Dept. (TEXT-MESS, TIN2006-15265-C06-05 and KNOW2, TIN2009-14715-C04-04). Daniel Ferrés was supported by a UPC-Recerca grant from Universitat Politècnica de Catalunya (UPC). TALP Research Center is recognized as a Quality Research Group (2001 SGR 00254) by DURSI, the Research Department of the Catalan Government.

References

1. Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J., Skalban, Y.: Getting Geographical Answers from Wikipedia: the GikiP pilot at CLEF. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, Springer, Heidelberg (2009)
2. Brants, T.: TnT – A Statistical Part-Of-Speech Tagger. In: Proceedings of the 6th Applied NLP Conference (ANLP 2000), Seattle, WA, United States (2000)
3. Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M.: FreeLing 1.3: Syntactic and Semantic Services in an Open-Source NLP Library. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), pp. 48–55 (2006)
4. Ferrés, D., Rodríguez, H.: TALP at GeoCLEF 2007: Results of a Geographical Knowledge Filtering Approach with Terrier. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 830–833. Springer, Heidelberg (2008)

Semantic QA for Encyclopaedic Questions: *EQUAL* in GikiCLEF

Iustin Dornescu

University of Wolverhampton, UK
I.Dornescu2@wlv.ac.uk

Abstract. This paper presents a new question answering (QA) approach and a prototype system, *EQUAL*, which relies on structural information from Wikipedia to answer open-list questions. The system achieved the highest score amongst the participants in the GikiCLEF 2009 task. Unlike the standard textual QA approach, *EQUAL* does not rely on identifying the answer within a text snippet by using keyword retrieval. Instead, it explores the Wikipedia page graph, extracting and aggregating information from multiple documents and enforcing semantic constraints. The challenges for such an approach and an error analysis are also discussed.

1 Motivation

This paper proposes a paradigm shift in open-domain question answering (QA) from the *textual approach*, built around a keyword-based document retrieval engine, to a *semantic approach*, based on concepts and relations. In its general form, open-domain QA is an AI-hard problem, causing research to focus on less complex types of questions which require a much smaller amount of world-knowledge. Evaluation fora such as TREC and QA@CLEF have focused on questions which can be answered by single textual snippets, such as definitions or factoid questions. As a consequence, most textual QA approaches are essentially smart paragraph retrieval systems, enhanced with extraction and ranking. Aggregating information from multiple documents usually means re-ranking candidates using their frequency-based confidence score. The textual approach has prevailed because computers do not have common-sense or reasoning abilities resembling those of humans. Standard textual QA systems do not 'understand' the contents of a textual document: they can only address questions which are answered by a snippet very similar to the question. While this approach works for some types of questions, it cannot extract answers when the supporting information is spread across different snippets from one or several documents.

The emergence of a web-of-data, the availability of linked-open-datasets and the amounts of information represented in ontologies, which allow a certain level of automated inference, prompt us to develop a new approach to QA: an architecture which is built around concepts rather than words in order to bridge the gap between the vast amounts of textual information available and the formal representations enabling automated inference.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 326–333, 2010.

This paper presents an alternative semantic QA architecture and prototype system, *EQUAL*, to address current limitations of textual QA. The rest of the paper is structured as follows: section 2 presents the key concepts in the proposed approach. The main processing steps employed by the prototype are outlined in section 3. Section 4 discusses the results and some limitations of the system. Section 5 comprises the error analysis. Section 6 describes relevant related work, and the paper finishes with conclusions and ideas for future work.

2 Semantic QA Architecture

To successfully address list questions, a novel architecture is proposed that does not have words at its core, but concepts, i.e. types (classes), entities (instances) and properties (relations and attributes). This allows the aggregation of information from across data sources, combining textual information with structured and semi-structured data in a way that enables basic inference in an open domain. The semantic QA architecture, of which *EQUAL* is a prototype, has two main processing phases: a) the **analysis** phase, which is responsible for understanding the question and identifying answers, and b) the **feedback** phase, responsible for interacting with the user. The latter is concerned with the efficient interface between the human and the machine, allowing, for example, disambiguation, justification and presentation of additional information using rich Web interfaces. Instead of 'guessing' the user's intent (as is the case for non-interactive QA), the system can provide visual feedback regarding its actions allowing the user to intervene during processing.

The data model adopted in the semantic QA architecture is closely related to RDF. Each entity corresponds to a node in a graph, described by attributes, such as *population size* or *date of birth*. Nodes are interlinked by relations such as *born in* or *president of*. This model facilitates the aggregation of information from various sources as needed at query time. The Wikipedia pages correspond to nodes, namely articles (entities, redirects, disambiguations and lists), categories (forming a folksonomy) and templates (primarily the infoboxes: semi-structured data). The text of an article is treated as an attribute in order to access information using NLP tools. The graph has links to open datasets such as dbpedia[1], Yago [9] and geonames[2] which provide additional attributes (e.g. type classification using WordNet nouns, population size, geographic coordinates).

By putting the concepts at the centre, a QA system can directly use non-textual resources from fields such as Information Extraction and Knowledge Representation, meaning that questions which have answers beyond the scope of a particular passage can be addressed. Systems should use structured data available from authoritative sources, and only extract information from text if necessary. In this unified approach, text is no longer a bag-of-words, some of which are Named Entities, i.e. location, person, company, but a context which refers to classes and instances, encoding their properties and their relations.

[1] http://wiki.dbpedia.org/About
[2] http://www.geonames.org/ontology/

During pre-processing, textual documents should be indexed based on unique identifiers of the mentioned entities, rather than on the words used. For example, the system should not retrieve passages containing the keyword *paris*, but the snippets referring to the French capital. References to other entities known as Paris (e.g. persons, other cities, films) should not be returned. When verifying the correctness of an answer, the system will examine the text from its page as well as text from the pages in its link neighbourhood, both those it refers to and those referring to it. The focus shifts from deducing information from small spans of text to accumulating and reusing world knowledge.

One of the key ideas behind the proposed architecture is its ability to address ambiguity. Rather than employing a pipeline of standard NLP tools to disambiguate the input and produce a most probable output, the system must be aware of different sources of ambiguity and be able to deal with them directly during processing. Humans use common sense and their knowledge of the world when understanding and interpreting questions. To compensate for the lack of such abilities, during the analysis phase the system must generate possible interpretations and rank them based on the results found and the context of the interaction. In the feedback phase, the system should be able to explain its 'understanding' of the question, and the criteria justifying the answers. The three challenges posed by this architecture are mapping natural language questions to semantic structures ('understanding' the question), representing available information according to the semantic model (retrieving the answers), and generating metadata to enable feedback and interaction.

3 *EQUAL*: A Prototype System for Encyclopaedic Question Answering for Lists

Based on the approach proposed above, the system developed for the GikiCLEF competition is a successor of the WikipediaListQA@wlv system [8]. Given that GikiCLEF is a non-interactive task, the current version of the prototype only implements the analysis phase.

The task of interpreting questions is difficult due to the ambiguity which characterizes natural language: very similar natural language expressions can have very different meanings while the same meaning can be expressed in a variety of forms. To tackle this problem, *EQUAL* tries to cover all the possible interpretations of a question, each corresponding to a different 'understanding'. An interpretation is a set of semantic constraints involving entities, relations and properties. The constraints used by the system are summarised below:

- **EAT** (expected answer type)**:** indicates the type of entities sought. Only entities of this type are considered valid answer candidates;
- **entity:** denotes an entity corresponding to a Wikipedia article;
- **relation:** indicates the relation between entities, represented by the verb;
- **property:** restricts a set of entities to a subset which have a certain property (e.g. *population size*);

- **geographic:** restricts the entities to having a relation with a particular geographic entity (e.g. *born in a country*, *contained in a region*);
- **temporal:** constrains the temporal interval of a relation;
- **introduction:** marks the phrase used to start the question and indicates redundancy. This chunk is removed from the question.

In the first processing step, *EQUAL* employs a chunking algorithm to decompose the question into constituents and determines the expected answer type. In general, this chunk is marked by an interrogative pronoun, but in the GikiCLEF collection, after the redundant introduction is removed, the first chunk always corresponded to the EAT constraint. This chunk is mapped to a Wikipedia category representing the set of valid candidate answers. The noun in a plural form determines the most generic mapping possible, while the remaining words are used to find a more specific (sub-)category.

In the second step, *EQUAL* creates interpretation objects by assigning a constraint type to each remaining chunk. The way these constraints are combined is related to the more general problem of parsing. As well as syntactic ambiguity, the system also deals with referential ambiguity, i.e. disambiguating eponymous entities, and type ambiguity, when a chunk could be interpreted as more than one type of semantic constraint. Figure 1 demonstrates the shallow parsing and semantic constraints for the first question in the test set.

Name || Romanian writers || who were living | in USA | in 2003
$\{_{intro}$Name$\}$ $\{_{eat}$Romanian writers$\}$ who were $\{_{rel}$living$\}$ $\{_{geo}$in USA$\}$ $\{_{temp}$in 2003$\}$

Fig. 1. Constraints in question GC-2009-01

The semantics of the constraints themselves in the context of Wikipedia are defined by *constraint verifiers*, i.e. the actual implementation which verifies whether a particular constraint holds. A constraint has several verifiers; these can be specialised for certain subsets of the data or can use external data sources.

In the third step, *EQUAL* explores the Wikipedia graph verifying which entities satisfy the constraints of the current question interpretation. *EQUAL* uses several constraint verifiers for the same semantic constraint, in order to take advantage of all the existing sources of information. For example, in Wikipedia, *geographic containment* can be expressed using demonym modifiers (*Nepalese*), categories (*Mountains of Nepal*), tables, infoboxes and text. The implemented verifiers use a set of generic patterns to process information from both the infoboxes and the article text (see [1]). An alternative would be to integrate a geographical Web-Service to verify relations such as containment, neighbouring and distance, using the geo coordinates provided by dbpedia. Extracting structured information from Wikipedia is an active area in the fields of Knowledge Representation and Information Extraction [4]. The challenge is to reuse and integrate resources developed in these domains.

EQUAL has an entity-centric design, given the article-entity duality in GikiCLEF. By abstracting questions as a set of constraints, the system is no

longer confined to the words of a particular snippet: when checking constraints for an entity, both the pages it refers to and those referring to it are examined. The prototype either extracts information from the page describing the subject of a triple: i.e. checking the page *Province of Florence* to see if it "produces" *Chianti*, or, symmetrically, from the page describing the object: i.e. *Chianti* to check if it is "produced" in the *Province of Florence*. More information regarding exploring the Wikipedia graph and verifying constraints is given in [1]. The general architecture allows for such a constraint to be validated from any source which mentions the two entities, including tables, list pages, navboxes, and perhaps wikified newswire articles [5,6].

4 Discussion and Results

For GikiCLEF 2009, the prototype implements a simplified version of the proposed architecture. This section presents the results and discusses some issues arising from this implementation, identifying solutions for future integration. The most important simplification is due to the non-interactive nature of the task. Lacking the feedback stage, *EQUAL* generates alternative question interpretations and addresses them sequentially, until one yields results. This is based on the hypothesis that only a correct interpretation of the question can give results. A better solution would be to rank the confidence of each answer set, but there was too little training data to create a reliable ranking measure.

The constraint verifiers employed are rather generic. Each has a method to be instantiated using a chunk from the question and a method to test a candidate solution. The implementations are quite general and offer a good balance between accuracy and coverage, but it is possible that more specialised implementations could give better performance. Given that the training set was relatively small, only a few verifiers were used to prevent over-fitting. A future challenge is to automatically learn verifiers from either seed examples or user feedback.

It is also computationally expensive to test all the verifiers at query time, especially those analysing the entire text of an article. Answering questions with a large number of candidate articles was aborted if no result was identified within a given time limit. To reduce the complexity, it is necessary to optimise the order of evaluating verifiers and extending the system to first use 'pre-verifiers', representing necessary conditions that can be tested efficiently against structured data. The time-consuming verifiers should be applied in a second step, which can be parallelised, to further reduce the answering time.

To deal with all ten languages of the GikiCLEF competition, *EQUAL* processed English due to the better coverage it affords. *EQUAL* returned 69 correct, 10 unjustified and 59 incorrect answers, giving a precision of 50% and a score of 34.5. By mapping these answers from English to the other nine languages using the interwiki links, the cumulative results total 385 correct answers out of 813: precision 47%, score 181.93.

The system ranked first in GikiCLEF, ahead of semi-automatic and standard textual QA approaches [7], proving that questions previously considered too

difficult to answer can be successfully addressed using Wikipedia as a repository of world-knowledge. As well as precision, the system also had the best performance in terms of the number of correct answers found. However, the results reflect the bias of the official scoring measure towards precision: the system did not return any answers for 25 (half) of the questions. In spite of the positive results, the prototype needs further improvements to be able to answer more questions and to increase its recall.

Table 1 shows the results without considering whether correct answers were justified or not. Relative recall (**rR**) is computed relative to the distinct correct answers in the English answer pool. The difference between the average precision per question and the overall precision reflects their opposite bias for questions with either few or many answers.

Table 1. Error analysis

Question Set	**avg. P**	**avg. rR**	**P** (overall)	**rR** (overall)
answered (25)	0.79	0.80	0.57 (79/138)	0.67 (79/118)
complete (50)	–	0.40	0.57 (79/138)	0.41 (79/192)

For the set of answered questions, performance is satisfactory despite most of *EQUAL*'s modules being relatively simple. While the prototype clearly outperformed standard textual QA systems [7], the competition uncovered some limitations of the current implementation. The fact that it did not return any answers for half of the questions suggests that some components have a limited coverage and prompts an analysis of the cause.

5 Error Analysis

The primary cause for imprecise answers is due to the fact that *EQUAL* uses few semantic constraints: it lacks the expressive power required to accurately represent all the questions. Sometimes, the generated interpretation misses relevant information from the question. For example, in GC-2009-11 *What Belgians won the Ronde van Vlaanderen exactly twice?*, *EQUAL* returns **all** the Belgian winners, because it cannot 'understand' the constraint *exactly twice*.

The verifiers are another cause of low performance. For example, in GC-2009-09 *Name places where Goethe fell in love*, the system cannot distinguish between the places where Goethe **lived**, and those where he **fell in love**. This is because the relation constraint verifier looks at the positions of the trigger words (*fell*, *love*) in relation to the link to a candidate answer. For topic GC-2009-19 *Name mountains in Chile with permanent snow*, the system only found one of the total of 14 answers judged correct, because its verifiers looked for an explicit reference to permanent snow. It is sometimes debatable what kind of

proof should be accepted as valid; for certain topics the judges had difficulties in reaching consensus, suggesting that the task is also difficult for humans.

The performance is also affected by inconsistencies in Wikipedia. *EQUAL* assumes that all the articles are assigned to correct and relevant categories, but this is not always the case. On the one hand, inaccurate categories decrease precision. For example, in GC-2009-18 *In which Tuscan provinces is Chianti produced?*, 13 pages were inaccurately assigned to the category *Provinces of Tuscany* at the time of the competition, when in fact they are places in Tuscany, and only 3 pages actually described provinces. On the other hand, missing categories translate to a decrease in recall. For example, in GC-2009-15 *List the basic elements of the cassata*, the question is asking for the **ingredients** of the Italian dessert (sponge cake, ricotta cheese, candied peel, marzipan, candied fruit, and so on), but none of them are part of an *ingredients* category, since it does not exist. Information in Wikipedia is continuously improved and updated, but such inconsistencies are inherent in a project this size [2].

Mapping the EAT to categories also needs further refinement, as the current mechanism assumes that there exists a most relevant category and that the terms in the questions are similar to Wikipedia's folksonomy. This is not always the case, usually because such a category does not exist (e.g. *German-speaking movies* and *Swiss casting show winners*). A bad category mapping is the primary cause for not answering questions: either the mapping is too generic, yielding too many results, or it is incorrect and no results are found. Given the bias towards precision in the official scoring, when *EQUAL* found more answers than the maximum threshold, the entire result-set was dismissed at the expense of recall.

6 Related Work

Extracting semantic information from Wikipedia is an active research area [4]. The most relevant IE system is WikiTaxonomy [10] which extracts explicit and implicit relations by splitting the category title. For example, from *Category:Movies directed by Woody Allen*, relations such as *directedBy* and *type(movie)* are extracted. *EQUAL* employs similar techniques to identify the best EAT category and to check geographic constraints.

Amongst QA sytems, InSicht [3] uses a semantic approach to QA which shares some similarities with *EQUAL*. It employs a decomposition technique for complex questions and uses several answer producers to boost recall. However, the semantic models used by two systems reside at different levels. InSicht uses a parser to transform both documents and questions into deep semantic representations, and matches these by drawing logical inferences at the snippet level. The core of the semantic representations used by *EQUAL* is instead basic world knowledge extracted from the Wikipedia link graph. The text is only preprocessed to test for the presence or absence of simple surface patterns. The two approaches have complementary strengths and could be combined, given that a successor of InSicht had encouraging results in GikiCLEF [7].

7 Conclusions and Future Work

This paper presented an architecture for semantic QA and a prototype system, *EQUAL*. At the core of the architecture is a set of semantic constraints which are used to represent the possible interpretations of questions. When extracting the answers for each interpretation, the constraints are verified using the underlying data and the set of available verifiers, and the candidate entities which satisfy all the constraints are selected. The system performed very well in GikiCLEF 2009, ranking first among 8 systems, proving adequate for the task. Further improvements are required to increase recall and to reduce the runtime complexity of the method.

Future work includes adding the feedback stage to increase user satisfaction. This type of interactive QA interface will allow the user to verify the criteria employed by the system and suggest changes or rephrase the question. The system can then learn from the choices made by its users. A suitable performance measure could be precision and recall of results found during a time-limited interactive session, to better resemble real-world usage of a QA system.

Acknowledgements. The development of the *EQUAL* system was partly supported by the EU-funded QALL-ME project (FP6 IST-033860).

References

1. Dornescu, I.: EQUAL: Encyclopaedic QUestion Answering for Lists. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for CLEF, Corfu, Greece (2009)
2. Hammwöhner, R.: Interlingual Aspects Of Wikipedia's Quality. In: Proc. of the Intl. Conf. On Information Qualiy - ICIQ (2007)
3. Hartrumpf, S., Glöckner, I., Leveling, J.: Efficient question answering with question decomposition and multiple answer streams. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 421–428. Springer, Heidelberg (2009)
4. Medelyan, O., Milne, D.N., Legg, C., Witten, I.H.: Mining meaning from Wikipedia. Intl. Journal of Human Computer Studies 67(9), 716–754 (2009)
5. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of CIKM 2007, pp. 233–242. ACM, New York (2007)
6. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of CIKM 2008, pp. 509–518. ACM, Napa Valley (2008)
7. Santos, D., Cabral, L.M.: GikiCLEF: Crosscultural Issues in an International Setting: Asking non-English-centered Questions to Wikipedia. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for CLEF, Corfu, Greece (2009)
8. Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J., Skalban, Y.: GikiP at GeoCLEF 2008: Joining GIR and QA Forces for Querying Wikipedia. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 894–905. Springer, Heidelberg (2009)
9. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: 16th Intl. WWW Conference, pp. 697–706. ACM Press, New York (2007)
10. Zirn, C., Nastase, V., Strube, M.: Distinguishing between Instances and Classes in the Wikipedia Taxonomy. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 376–387. Springer, Heidelberg (2008)

Interactive Probabilistic Search for GikiCLEF

Ray R. Larson

School of Information
University of California, Berkeley, USA
ray@sims.berkeley.edu

Abstract. In this paper we will briefly describe the approaches taken by the Berkeley Cheshire Group for the GikiCLEF task of the QA track. Because the task was intended to model some aspects of user search, and because of the complexity of the topics and their geographic elements, we decided to conduct interactive searching of the topics and selection of results. Because of the vagueness of the task specification early-on, some disagreements about what constituted a correct answer, and time constraints we were able to complete only 22 of the 50 topics. However, in spite of this limited submission the interactive approach was very effective and resulted in our submission being ranked third overall in the results.

1 Introduction

We began the GikiCLEF task in somewhat of a quandary. In past CLEF tasks we had relied entirely on machine translation tools and fully automatic search methods. But it was clear from the GikiCLEF task description[4] that some form of interactive search was intended, and that the topics would be much more involved and complex than previous CLEF topics, although the original task description on the GikiCLEF web site is not at all clear on what constitutes an "answer" to a particular question. Because we did not know enough about the task to attempt to construct a fully automated approach, we decided instead to construct an interactive IR system that was able to search across all of the Wikipedia test collections in each of the target languages (Bulgarian, Dutch, English, German, Italian, Norwegian (bokmaal), Norwegian (nynorsk), Portuguese, Romanian and Spanish)

What was not clear was that, unlike all of the other CLEF tasks, the intended answers to the questions could *not just be passages in the text of the articles* but had to be exactly the *title of the article*, and that title had to be of a specific type (often a place) that was supposed to be inferred from the form of the question. This constraint effectively eliminated the possibility for fully automatic methods (at least given the techniques we had readily available), so we decided to use this first participation in GikiCLEF as an exploratory study of what kinds of search might prove useful in this task

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 334–341, 2010.

In this paper we will very briefly discuss the retrieval algorithms employed in our interactive system, provide some description of the system itself, and offer some comments on the evaluation and various issues that arose. Finally we discuss the barriers to effective automatic processing of the GikiCLEF task.

2 The Retrieval Algorithms

The retrieval algorithms used for our GikiCLEF interactive system include ranked retrieval using our Logistic Regression algorithms combined with Boolean constraints, as well as simple Boolean queries for link following, etc. Because the basic form and variables of the *Logistic Regression* (LR) algorithm used for all of our submissions was virtually identical to one that appears in our papers from previous CLEF participation[3,2], so the details will be omitted here. We used both the "TREC2" and the "TREC3" algorithms in this task, along with a probabilistically based "psuedo" or "blind" relevance feedback in tandem with the TREC2 algorithm. These algorithms (and especially the TREC2 with blind relevance feedback) have performed well in a variety of IR tasks at CLEF and other IR evaluations [1]. The basic form of the logistic regression algorithm is, in effect, an estimated model for relevance prediction:

$$\log O(R \mid Q, D) = b_0 + \sum_{i=1}^{S} b_i s_i \tag{1}$$

where b_0 is the intercept term and the b_i are the coefficients obtained from the regression analysis of a sample collection and relevance judgements (the algorithm we used was trained on TREC data). Each of the s_i is a collection, topic, or document statistic (such as document term frequency). The statistics used are based on the usual measures such as term frequency, inverse document frequency, etc. that seem to be part of most IR algorithms. The final ranking is determined by the conversion of the log odds form to probabilities:

$$P(R \mid Q, D) = \frac{e^{\log O(R|Q,D)}}{1 + e^{\log O(R|Q,D)}} \tag{2}$$

This step is not really necessary, since the probabilities and log odds would have the same order in a ranking, but we use it to combine with other probabilistic calculations.

2.1 Boolean Search Operations

To enable efficient browsing in an interactive system it is also useful to implement Boolean constraints and search options. These are built into the Cheshire II system and use the same indexes as the ranked search operations. For this implementation we typically used a Boolean AND search of all of the query words combined by Boolean AND with the results of a ranked search of those words.

Table 1. Cheshire II Indexes for GikiCLEF 2009

Name	Description	Content Tags
title	Item Title	title tag
meta	Content Metadata	content attribute of meta tag
topic	Most of Record	title, body and meta@content tags
anchors	Anchor text	anchor (a) tags

The final estimate for the estimated probability of relevance used for ranking the results of a search combining Boolean and logistic regression strategies is simply:

$$P(R \mid Q, D) = P(R \mid Q_{bool}, D)P(R \mid Q_{prob}, D) \tag{3}$$

where $P(R \mid Q_{prob}, D)$ is the probability estimate from the probabilistic portion of the search, and $P(R \mid Q_{bool}, D)$ the estimate from the Boolean, which will be either 0 or 1 depending on whether the Boolean constraint does not or does match the document. For constraints that require all terms in the query to be in the document this operation retains the ranking values generated by the ranked search while limiting the results to only those that contain all of the terms.

In addition, to implement the internal links of the Wikipedia test collections for the interactive system, each link in a retrieved page was converted to a Boolean title search for the linked page name. Thus, instead of following links directly each link became a search on a title. Direct use of the links was impossible since the collection pages were not preserved with in the same file structure as the original Wikipedia and names in the links and the actual page file names differed due to additions during the collection creation process.

Also a Boolean AND NOT search was used to help filter results in some cases with ambiguous terms.

3 Approaches for GikiCLEF

In this section we describe the specific approaches taken for our submitted runs for the GikiCLEF task. First we describe the indexing and term extraction methods used, and then the search features we used for the submitted runs.

3.1 Indexing and Term Extraction

The Cheshire II system uses the XML (in this case the XHTML) structure of the documents to extract selected portions for indexing and retrieval. Any combination of tags can be used to define the index contents.

Table 1 lists the indexes created by the Cheshire II system for each language collection of the GikiCLEF Wikipedia database and the document elements from which the contents of those indexes were extracted. For each of the languages:

Bulgarian, Dutch, English, German, Italian, Norwegian (bokmaal), Norwegian (nynorsk), Portuguese, Romanian and Spanish, we tried to use, where possible, language-specific stemmers and stoplists whenever possible. Our implementation of the Snowball stemmer is some years old and lacked stemmers for Bulgarian, Norwegian(bokmaal) and Romanian. For these we substituted a stemmer with somewhat similar language roots. I.e., a Russian stemmer for Bulgarian, Norwegian(nyorsk) for Norwegian(Bokmaal) and Italian for Romanian.

For all indexing we used language-specific stoplists to exclude function words and very common words from the indexing and searching. The German language runs *did not* use decompounding in the indexing and querying processes to generate simple word forms from compounds. The Snowball stemmer was used by Cheshire for language-specific stemming.

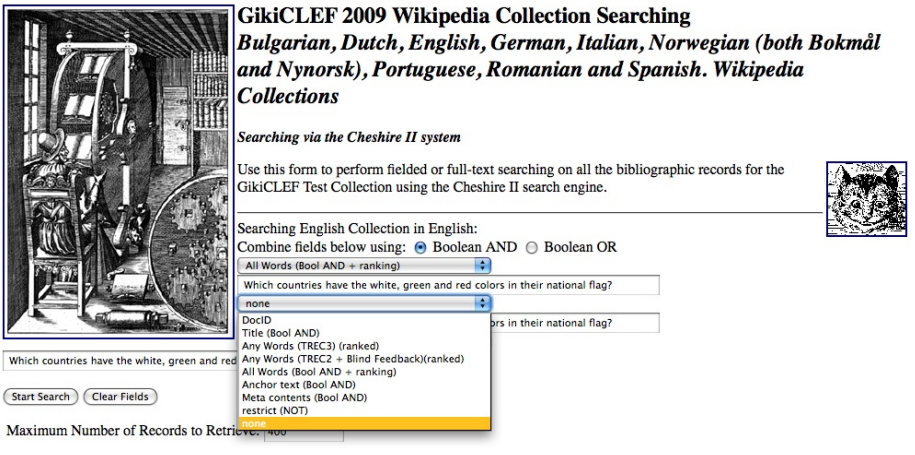


Fig. 1. Search Form in the Interactive Search System

3.2 Search Processing

Interactive searching of the GikiCLEF Wikipedia collections used the Cheshire II system via a set of web pages and TCL scripts that allowed the searcher to select a particular topic id and language and have it loaded into a search form for manual modification and selection of search indexes and approaches. Figure 1 shows this form for topic GC-2009-02. Typically the user would edit the query to remove extraneous terms, and submit the query, leading to a ranked result list page (Figure 2). From the ranked result list page the user can click on the article title to see the full page (Figure 3) or click on any of the language codes on the line to submit that title as title query in the Wikipedia collection for that language (the user is also given a chance to edit the search before it is submitted to allow language-specific adaptations). From a page like that shown in Figure 3, any of the links can be clicked on generating a Boolean title search for that page.

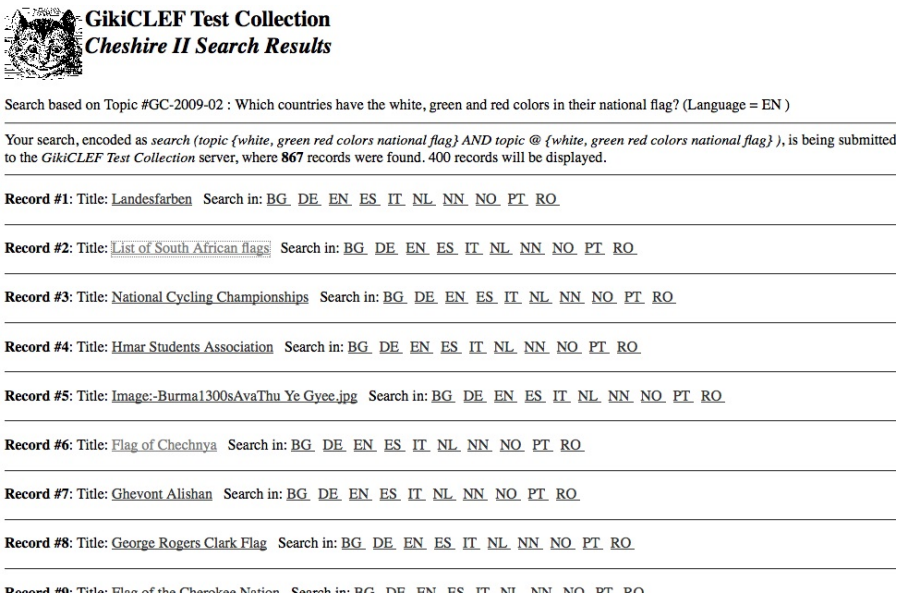


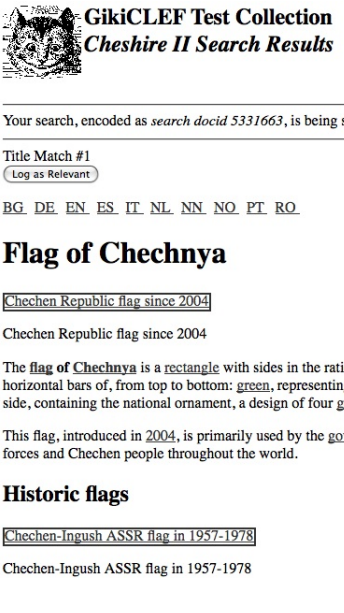
Fig. 2. Ranked List of Results

For example clicking on the country name link "Chechnya" in the first line leads to a list of pages containing the word "Chechnya" in their titles, one of which is the specific country page shown in Figure 4.

Each display of a full page includes a "Log as Relevant" button to save the page information in a log file. This log file is the basis for the submitted results for the GikiCLEF task.

4 Results for Submitted Runs

Needless to say, doing the GikiCLEF task interactively involved a lot of time spent reading pages and deciding whether or not the page was relevant. The author was the only person participating in this evaluation, so all system development choices and query choices were my own. As it turned out in the evaluation many of the pages that I believed to be relevant (such as the page shown in Figure 3 were judged not to be relevant.) Although in this particular case it is very difficult to understand why, for the topic "Which countries have the white, green and red colors in their national flag?" the article entitled "Flag of Chechnya" is considered NOT relevant while the article "Chechnya" IS (even though the colors of the flag are never mentioned and no images were included in the collections). The official position is that the question was about



GikiCLEF Test Collection
Cheshire II Search Results

Your search, encoded as *search docid 5331663*, is being submitted to the *GikiCLEF Test Collection* server, where **1** record was found.

Title Match #1
Log as Relevant

BG DE EN ES IT NL NN NO PT RO

Flag of Chechnya

Chechen Republic flag since 2004

Chechen Republic flag since 2004

The **flag of Chechnya** is a rectangle with sides in the ratio 2:3, the same ratio as the flag of the Russian Federation. The flag is composed of three horizontal bars of, from top to bottom: green, representing Islam; white; and red; superimposed on them is a narrow vertical white band at the hoist side, containing the national ornament, a design of four golden scroll shapes.

This flag, introduced in 2004, is primarily used by the government of Chechnya while the independentist flags are commonly used by opposition forces and Chechen people throughout the world.

Historic flags

Chechen-Ingush ASSR flag in 1957-1978

Chechen-Ingush ASSR flag in 1957-1978

Fig. 3. Search Result with Language Search Links

REDIRECT TO: Chechnya

Title Match #26
Log as Relevant

BG DE EN ES IT NL NN NO PT RO

Chechnya

Chechen Republic (English)
Чеченская Республика (Russian)
Нохчийн Республика (Chechen)

Location of the Chechen Republic in Russia

Coat of Arms	Flag
Coat of arms	border Flag of Chechnya

Anthem: Anthem of the Chechen Republic

Capital	Grozny
Established	January 11 , 1991
Political status	Republic
Federal district	Southern
Economic region	North Caucasus
Code	20
Area	
Area - Rank within Russia	15,300 km² 75th
Population	

Fig. 4. Multilingual Results with Various Search Links

the country, and therefore country names alone are acceptable (the fact that the country name is ALSO included in the non-relevant "Flag" item does not seem to matter).

In any case, because each question took literally hours of work using the interactive system, and my time was constrained by other considerations I completed and submitted only 22 out of the 50 topics, with results from all of the target languages of the collections.

As the results table in the overview paper for the GikiCLEF track[4] shows that the interactive approach was fairly effective in spite of not completing all of the topics (our scores are labeled "Cheshire").

5 Conclusions

In looking at the overall results for the various GikiCLEF tasks, it would appear that the interactive approach using logistic regression ranking Boolean constraints can provide fairly good results. But what would be required for completely automatic processing with comparable results? For each of the topics provided for searching, the terms that might lead to effective search are combined with terms that are much more difficult to deal with automatically. These words are not really candidates for a stoplist, since in other contexts they might be quite effective search terms, but their effective use requires that 1) full high-quality NLP part of speech tagging should be performed on the topics, 2) question (and result) type information be inferred from the topic, and 3) results be categorized as to their type in order to match with the type inferred. None of these are trivial tasks and inference required may well be beyond the capabilities of current NLP tools. Many, if not most of the topics have multiple facets, but usually only responses to a particular single facet are considered correct. At times determining this facet may difficult even for a human searcher, much less an automated system. Consider topic #6 "Which Dutch violinists held the post of concertmaster at the Royal Concertgebouw Orchestra in the twentieth century?". From this we are meant to infer that only citizens or people born in the Netherlands are valid answers, moreover they must play or have played the violin, and they must have become concertmasters at the Royal Concertgebouw Orchestra, and moreover this position was held during the twentieth century. Without satisfying all of these facets, the answer cannot be correct (E.g. A Belgian violist as concert master would be wrong, as would a Dutch violist as concert master if he or she was concertmaster in the 1870's.

Since GikiCLEF is a new task for us, we took a fairly conservative approach using methods that have worked well in the past, and used our interaction with the collection to try to discover how this kind of searching might be implemented automatically. There are no simple answers for this task with its complex questions and constraints, but through our interactive work we think we have some possible strategies for future evaluation.

References

1. Larson, R.R.: Probabilistic retrieval, component fusion and blind feedback for XML retrieval. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 225–239. Springer, Heidelberg (2006)
2. Larson, R.R.: Cheshire at geoclef 2007: Retesting text retrieval baselines. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 811–814. Springer, Heidelberg (2008)
3. Larson, R.R.: Experiments in classification clustering and thesaurus expansion for domain specific cross-language retrieval. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 188–195. Springer, Heidelberg (2008)
4. Santos, D., Cabral, L.M.: Gikiclef: Expectations and lessons learned. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 212–222. Springer, Heidelberg (2010)

Information Filtering Evaluation: Overview of CLEF 2009 INFILE Track

Romaric Besançon[1], Stéphane Chaudiron[2], Djamel Mostefa[3], Ismaïl Timimi[2], Khalid Choukri[3], and Meriama Laïb[1]

[1] CEA LIST, 18, route du panorama BP 6 92265 Fontenay aux Roses France
[2] Université de Lille 3 – GERiiCO Domaine univ. du Pont de Bois 55-57, BP 60149 - 59653 Villeneuve d'Ascq cedex France
[3] ELDA rue Brillat Savarin 75013 Paris France
romaric.besancon@cea.fr, meriama.laib@cea.fr, stephane.chaudiron@univ-lille3.fr, mostefa@elda.org, ismail.timimi@univ-lille3.fr, choukri@elda.org

Abstract. The INFILE@CLEF 2009 is the second edition of a track on the evaluation of cross-language adaptive filtering systems. It uses the same corpus as the 2008 track, composed of 300,000 newswires from Agence France Presse (AFP) in three languages: Arabic, English and French, and a set of 50 topics in general and specific domains (scientific and technological information). In 2009, we proposed two tasks : a batch filtering task and an interactive task to test adaptive methods. Results for the two tasks are presented in this paper.

1 Introduction

The purpose of the INFILE (INformation FILtering Evaluation) track is to evaluate cross-language adaptive filtering systems. The goal of these systems is to successfully separate relevant and non-relevant documents in an incoming stream of textual information with respect to a given profile. The document and profile being possibly are written in different languages.

The INFILE track has first been run as a pilot track in CLEF 2008 campaign [1]. Due to some delays in the organization, the participation in the 2008 was weak (only one participant submitted results), so we decided to propose to rerun the campaign in 2009, using the same document collection and topics. It would have been better to use a brand new evaluation set, with new documents, new topics and new judgments but this is a heavy and expensive task, especially since we are dealing with multilingual comparable documents and topics. Since there was only one participant in 2008, we decided to run the evaluation campaign with the same data.

The INFILE track is originally funded by the French National Research Agency and co-organized by the CEA LIST, ELDA and the University of Lille3-GERiiCO.

In this track, information filtering is considered in the context of competitive intelligence and the evaluation protocol of the campaign has been designed with a particular attention to the use of filtering systems by real professional users. Even if the

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 342–353, 2010.

campaign is mainly a technological oriented evaluation process, we adapted the protocol and the metrics, as close as possible, to how a user would proceed, including through some interaction and adaptation of his system.

The INFILE campaign can mainly be seen as a cross-lingual pursuit of the TREC 2002 Adaptive Filtering task [2] (adaptive filtering track has been run from 2000 to 2002), with a particular interest in the correspondence of the protocol with the ground truth of competitive intelligence (CI) professionals. In this goal, we asked CI professionals to write the topics according to their experience in the domain.

Other related campaigns are the Topic Detection and Tracking (TDT) campaigns from 1998 to 2004 [3], but in the TDT campaigns, focus was mainly on topics defined as "events", with a fine granularity level, and often temporally restricted, whereas in INFILE (similar to TREC 2002), topics are of long-term interest and supposed to be stable, which can induce different techniques, even if some studies show that some models can be efficiently trained to have good performance on both tasks [4].

2 Description of the Tasks

In addition to the adaptive filtering task already proposed in 2008 [1], we introduced the possibility to test batch filtering systems in 2009. For both tasks, the document collection consists in a set of newswire articles provided by the Agence France Presse (AFP) covering recent years. The topic set is composed of two different kinds of topics, one concerning general news and events, and a second one on scientific and technological subjects.

The filtering process may be crosslingual: English, French and Arabic are available for the documents and topics, and participants may be evaluated on monolingual runs, bilingual runs, or multilingual runs (with several target languages).

The purpose of the information filtering process is to associate documents in an incoming stream to zero, one or several topics: filtering systems must provide a Boolean decision for each document with respect to each topic. The evaluation corpus consisted of 300,000 documents (100,000 documents per language).

For the batch filtering task, participants are provided with the whole document collection and must return the list of relevant documents for each topic (since the filtering process supposes a binary decision for each document, the document list does not need to be ranked).

For the adaptive filtering task, the evaluation is performed using an automatic interactive process, with a simulated user feedback: for each document considered relevant to a topic, systems are allowed to ask for a feedback on this decision (i.e. ask if the document was indeed relevant for the topic or not), and can modify their behavior according to the answer. The feedback is allowed only on kept documents, there is no relevance feedback possible on discarded documents. In order to simulate the limited patience of the user, a limited number of feedbacks is allowed: this number has been fixed in 2009 to 200 feedbacks by run (it was 50 in 2008; but most participants considered this insufficient).

The adaptive filtering task uses an interactive client-server protocol, that is described in more detail in [1]. The evaluation worked as follows:

1. the system registers to the document server
2. it retrieves a document Di from the server
3. it compares the document to each topic. For each topic Tj for which the document Di is relevant it sends the pair (Tj, Di) to the server
4. for each pair (Tj, Di) the server can ask for feedback; the server returns a Boolean answer indicating if the association (Tj, Di) is correct or not. The number of feedbacks is limited to 200; after this number is reached the server returns always false.
5. A new document can be retrieved (back on 2).

The batch filtering task has been run from April 2nd (document collections and topics made available to the participants) to June 1st (run submission), and the adaptive filtering task has been run from June 3rd to July 10th.

3 Test Collections

Topics. A set of 50 topics (or profiles) has been prepared, covering two different categories: the first group (30 topics) deals with general news and events concerning national and international affairs, sports, politics etc and the second one (20 topics) deals with scientific and technological subjects. The scientific topics were developed by CI professionals from INIST[1], ARIST Nord Pas de Calais[2], Digiport[3] and OTO Research[4]. The topics were developed in both English and French. The Arabic version has been translated from English and French by native speakers.

Topics are defined with the following elements: a unique identifier, a title (6 full words max.) describing the topic in a few words, a description (20 words max.) corresponding to a sentence-long description, a narrative (60 words max.) corresponding to the description of what should be considered a relevant document and possibly what should not, keywords (up to 5) and an example of relevant text (120 words max.) taken from a document that is not in the collection (typically from the web).

Each record of the structure in the different languages corresponds to translations, except for the samples which need to be extracted from real documents. An example of topic in the three languages is presented in Fig. 1.

Documents. The INFILE corpus is provided by the Agence France Presse (AFP) for research purposes. We used newswire articles in 3 languages: Arabic, English and French[5] covering a three-year period (2004-2006) which represents a collection of about one and half million newswires for around 10 GB, from which 100,000 documents of each language have been selected to be used for the INFILE test.

[1] The French Institute for Scientific and Technical Information Center, http://international.inist.fr/

[2] Agence Régionale d'Information Stratégique et Technologique, http://www.aristnpdc.org/

[3] http://www.digiport.org

[4] http://www.otoresearch.fr/

[5] Newswires in different languages are not translations from a language to another (it is not an aligned corpus): the same information is generally rewritten to match the interest of the audience in the corresponding country.

<top>
<num>147</num>
<title>Care management of Alzheimer disease</title>
<desc>News in the care management of Alzheimer disease by families, society and politics</desc>
<narr>Relevant documents will highlight differents aspects of Alzheimer disease management: - human involvement of carers : families, health workers - financial means: nursing facilities, diverse grants to carers - political decisions leading to guidelines for optimal management of this great public health problem </narr>
<keywords>
<k>Alzheimer disease</k>
<k>Dementia </k>
<k>Care management </k>
<k>Family support </k>
<k>Public health</k>
</keywords>
<sample>The AAMR/IASSID practice guidelines, developed by an international workgroup, provide guidance for stage–related care management of Alzheimer's disease, and suggestions for the training and education of carers, peers, clinicians and programme staff. The guidelines suggest a three-step intervention activity process, that includes: (1) recognizing changes; (2) conducting...</sample>
</top>

<top>
<num>147</num>
<title>Prise en charge de la maladie d'Alzheimer</title>
<desc>Actualités dans le domaine de la prise en charge de la maladie d'Alzheimer, tant au niveau des familles, de la société qu'au niveau des choix politiques</desc>
<narr>Les documents pertinents présenteront les divers aspects de la prise en charge de la maladie d'Alzheimer : - moyens humains mis en jeu : familles, personnels de santé - moyens financiers : structures d'accueil, aides diverses aux malades et aux aidants - décisions politiques avec établissement de recommandations permettant d'encadrer de façon optimale ce problème majeur de santé publique </narr>
<keywords>
<k>Maladie d'Alzheimer</k>
<k>Démence </k>
<k>Prise en charge </k>
<k>Aide aux familles </k>
<k>Santé publique </k>
</keywords>
<sample>Un an après l'entrée en vigueur du plan ministériel, un rapport de l'OPEPS rendu public le 12 juillet 2005 dresse un bilan assez sévère de la prise en charge de la maladie d'Alzheimer et des maladies apparentées. Selon l'OPEPS*, la politique de prévention des facteurs de risque est insuffisante, ... </sample>
</top>

<top>
<num>147</num>
<title>العناية بمرض الزهايمر</title>
<desc>الأحداث المتعلقة بالعناية بمرض الزهايمر، على مستوى الأسر والمجتمع وأيضا على مستوى الاختيارات السياسية.</desc>
<narr>الوثائق التي تتعلق بالعناية بمرض الزهايمر من مختلف الجوانب : - الإمكانات البشرية المستخدمة : الأسر، موضفو الصحة، - الموارد المالية : بنيات الإستقبال، المساعدات المختلفة للمرضى والمساعدين، - القرارات السياسية : التعليمات الصادرة من أجل وضع إطار أمثل لهذا المشكل الكبير في الصحة العمومية.</narr>
<keywords>
<k>الصحة العمومية</k>
<k>مساعدة الأسر</k>
<k>عناية</k>
<k>الجنون</k>
<k>مرض الزهايمر</k>
</keywords>
<sample>... الوضع عبر الهاتف كلما اقتضت الحاجة ذلك. وكانت دراسة سابقة قد كشفت أن عدد المصابين بمرض الزهايمر سيتضاعف أربع مرات خلال العقود الأربعة المقبلة، ويصيب واحدا من أصل كل 85 شخصاً على وجه الأرض.وأكدت الدراسة أن هذه الإحصائية المخيفة مرتبطة بشكل رئيسي بارتفاع عدد كبار السن في مختلف دول العالم، الناجم عن تحسن الأنظمة الصحية، وقدرت أنه بحلول العام 2050 فإن أعداد أولئك المرضى ستقفز إلى 62.8 مليون شخص. بحسب CNN.</sample>
</top>

Fig. 1. An example of topic for the INFILE track, in the three languages

News articles are encoded in XML format and follow the News Markup Language (NewsML) specifications[6]. An example of document in English is given in Fig. 2. All fields are available to the systems and can be used in the filtering process (including keywords, categorization...).

Since we need to provide a real-time simulated feedback to the participants, we need to have the identification of relevant documents prior to the campaign, as in [5]. The method used to build the collection of documents with the knowledge of the relevant documents is presented in detail in [1]. A summary of this method is given here.

We used a set of 4 search engines (Lucene[7], Indri[8], Zettair[9] and the search engine developed at CEA-LIST) to index the complete collection of 1.4 million documents. Each search engine has been queried using different fields of the topics, which provides us with a pool of runs. We first selected the first 10 retrieved documents of each run, and these documents were assessed manually. We then iterate using a *Combination of Experts* model [6], computing a score for each run according to the current assessment and using this score to weight the choice of the next documents to assess. The final document collection is then built by taking all documents that are relevant to at least one topic (core relevant corpus), all documents that have been

[6] NewsML is an XML standard designed to provide a media-independent, structural framework for multi-media news. NewsML was developed by the International Press Telecommunications Council. see `http://www.newsml.org/`

[7] `http://lucene.apache.org`

[8] `http://www.lemurproject.org/indri`

[9] `http://www.seg.rmit.edu.au/zettair`

```
<NewsML Version="1.1">
 <NewsEnvelope>
  <TransmissionId>807</TransmissionId>
  <DateAndTime>20050615T212137Z</DateAndTime>[...]
 </NewsEnvelope>
 <NewsItem>
  <Identification>
   <NewsIdentifier>
    <ProviderId>afp.com</ProviderId>
    <DateId>20050615</DateId>
    <NewsItemId>TX-SGE-DPE59</NewsItemId>
    <RevisionId PreviousRevision="0" Update="N">1</RevisionId>
    <PublicIdentifier>urn:newsml:afp.com:20050615:TX-SGE-DPE59:1</PublicIdentifier>
   </NewsIdentifier>
   <NameLabel>Mideast-unrest-Israel-Palestinians</NameLabel>
  </Identification>
  <NewsManagement>[...]</NewsManagement>
  <NewsComponent>
   <TopicSet FormalName="NewsTopics">
    <Topic Duid="topic1"><TopicType FormalName="SlugKeyword"/><Description>Mideast</Description></Topic>
    <Topic Duid="topic2"><TopicType FormalName="SlugKeyword"/><Description>unrest</Description></Topic>
    <Topic Duid="topic3"><TopicType FormalName="SlugKeyword"/><Description>Israel</Description></Topic>
    <Topic Duid="topic4"><TopicType FormalName="SlugKeyword"/><Description>Palestinians</Description></Topic>
   </TopicSet>
   <NewsLines>
    <SlugLine>Mideast-unrest-Israel-Palestinians</SlugLine>
    <HeadLine>Israel says teenage would-be suicide bombers held</HeadLine>
   </NewsLines>
   <AdministrativeMetadata>[...]</AdministrativeMetadata>
   <DescriptiveMetadata>
    <Language FormalName="en"/>
    <SubjectCode><Subject FormalName="11999000"/></SubjectCode>
    <SubjectCode><Subject FormalName="INT" Vocabulary="urn:newsml:afp.com:20011001:AFPCatCodes:1"/></SubjectCode>
    <Location>
     <Property FormalName="Country" Value="ISR"/>
     <Property FormalName="City" Value="JERUS"/>
    </Location>
   </DescriptiveMetadata>
   <ContentItem>
    <MediaType FormalName="Text"/>
    <Format FormalName="NITF3.1-body.content"/>
    <Characteristics><Property FormalName="Words" Value="89"/></Characteristics>
    <DataContent>
     <p>JERUSALEM, June 15 (AFP) - The Israeli security service said Wednesday it had arrested four Palestinian teenage boys who
were preparing to carry out suicide bombings.Shin Beth said the four, aged 16 and 17, belonged to the Fatah movement. It said they planned
to hit targets in Israel or Israeli troops.</p>
     <p>Four other young adults, also accused of Fatah membership, were picked up in Nablus in the north of the West Bank some
weeks ago.</p>
     <p>Shin Beth said the network was financed by the Shiite Lebanese Hezbollah group.</p>
     <p>ms/sj/gk</p>
    </DataContent>
   </ContentItem>
  </NewsComponent>
 </NewsItem>
</NewsML>
```

Fig. 2. An example of document in the INFILE collection, in NewsML format

assessed and judged not relevant (difficult corpus: documents are not relevant, but share something in common with at least one topic, since they have been retrieved by at least one search engine), and a set of documents taken randomly in the rest of the collection (filler corpus, with documents that have not been retrieved by any search engines for any topic, which should limit the number of relevant documents in the corpus that have not been assessed).

Statistics on the number of assessed documents and relevant documents is presented in Table 1. We notice a difference between languages in terms of number of relevant documents: most topic designers were French speakers with English skills and the topics have been designed by exploring the French/English part of the document collection, which can explain this bias (more in French and English than in Arabic).

Table 1. Statistics on the number of assessed documents and the number of relevant documents, in each language

	eng	fre	ara
number of documents assessed	7312	7886	5124
number of relevant documents	1597	2421	1195
avg number of relevant docs / topic	31.94	48.42	23.9
std deviation on number of relevant docs / topic	28.45	47.82	23.08
[min,max] number of relevant docs / topics	[0,107]	[0,202]	[0,101]

The repartition of relevant documents across topics presented in Fig. 3 illustrates the difference of volume of data between topics, and show also that topics tend to have the same difficulty across languages (topics with few relevant documents in general will have few relevant documents in each language).

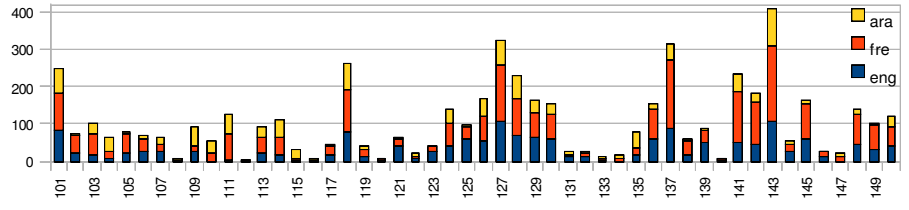


Fig. 3. Number of relevant documents for each topic, in each language

4 Metrics

The results returned by the participants are binary decisions on the association of a document with a topic. The results, for a given topic, can then be summarized in a contingency table of the form:

	Relevant	Not Relevant
Retrieved	a	b
Not Retrieved	c	d

On these data, a set of standard evaluation measures is computed:

- Precision, defined as
$$P = a/a+b \quad (1)$$
- Recall , defined as
$$R = a/a+c \quad (2)$$
- F-measure, which is a combination of precision and recall [7]. We used the standard combination that gives the same importance to precision and recall, defined by
$$F = 2\times PR/P+R \quad (3)$$

Following the TREC Filtering tracks [8,2] and the TDT 2004 Adaptive tracking task [3], we also consider the linear utility, defined as

$$u = w_1 \times a - w_2 \times b \tag{4}$$

where w_1 is the importance given to a relevant document retrieved and w_2 is the cost of an non relevant document retrieved.

Linear utility is bounded positively, but unbounded negatively (negative values depend on the number of relevant documents for a topic). Hence, the average value on all topics would give too much importance to the few topics on which a systems would perform poorly. To be able to average the value, the measure is scaled as follows:

$$u_n = \frac{\max(u/u_{max}, u_{min}) - u_{min}}{1 - u_{min}} \tag{5}$$

where u_{max} is the maximum value of the utility and u_{min} a parameter considered to be the minimum utility value under which a user would not even consider the following documents for the topic. In the INFILE campaign, we used the values $w_1 = 1$, $w_2 = 0.5$, $u_{min} = -0.5$ (same as in TREC 2002).

We considered in 2008 the *detection cost* measure (from the Topic Detection and Tracking campaigns [9]), but we do not present this score in this paper (we found that detection cost values were often low and not really discriminant between participants).

To compute average scores, the values are first computed for each topic and then averaged (*i.e.*, we consider macro-averaged scores). In order to measure the impact of the feedback and the adaptivity of the systems in the adaptive filtering track, the measures are also computed at different times in the process, each 10,000 documents, to get an evolution curve of the different values across time.

Additionally, we use the two following measures, introduced in the first INFILE campaign [1]: the first one is an originality measure, defined by the number of relevant documents the system uniquely retrieves (*i.e.* the number of relevant documents retrieved by the system and not retrieved by the other participants). It is designed as a comparative measure to give more importance to systems that use innovative and promising technologies that retrieve "difficult" documents (*i.e.* documents that are not generally retrieved by the participants, which are supposed to share enough common features with the topic to be considered as "easy").

The second one is an anticipation measure, designed to give more interest to systems that can find the first document in a given topic. This measure is motivated in CI by the interest of being at the cutting edge of a domain, and not missing the first information to be reactive. It is measured by the inverse rank of the first relevant document detected (in the list of documents), averaged on all topics. The measure is similar to the mean reciprocal rank (MRR) used for instance in Question Answering Evaluation [10], but is not computed on the ranked list of retrieved documents but on the chronological list of relevant documents.

5 Overview of the Results

Five participants (out of 9 registered) submitted results: 3 participants submitted results for the batch filtering task (a total of 9 runs), 2 for the interactive filtering task (3 runs). Participants were different for the two tasks. The participant are presented in Table 2, and the characteristics of the runs are detailed in Table 3.

Table 2. List of participants

team name	institute	country
IMAG	Institut Informatique et Mathématiques Appliquées de Grenoble	France
SINAI	University of Jaen	Spain
UAIC	Universitatea Alexandru Ioan Cuza of IASI	Romania
HossurTech	société CADEGE	France

Concerning the languages, 6 runs out of 9 are monolingual English for the batch filtering task, 3 are multilingual from English to French/English. For the interactive task, one run is monolingual English, one is monolingual French, and one is bilingual French to English. Unfortunately, no participant submitted runs with Arabic as source or target language (most participants participated for the first time and wanted to test their filtering strategies on more simple languages).

Table 3. The runs, by team and by run name, and their characteristics

team	run	task	source	target
IMAG	IMAG_1	batch	eng	eng
IMAG	IMAG_2	batch	eng	eng
IMAG	IMAG_3	batch	eng	eng
UAIC	uaic_1	batch	eng	eng
UAIC	uaic_2	batch	eng	eng-fre
UAIC	uaic_3	batch	eng	eng-fre

Participants used several approaches to tackle the different issues of the track: concerning the technologies used for the filtering itself, UAIC used an adapted IR tool (Lucene), SINAI used a SVM classifier, with the possibility to learn from external resources (Google), IMAG and HossurTech used textual similarity measures between topics and documents with selection thresholds to accept or reject the document and UOWD used a reasoning model based on the Human Plausible Reasoning theory. Concerning the adaptativity challenge, HossurTech used an automated variation of the selection threshold, and UOWD integrated the user feedback as a parameter of the reasoning model. Finally, to deal with crosslingual runs, the participants used bilingual dictionaries or machine translation techniques.

Evaluation scores[10] for all runs are presented in Table 4, for batch filtering (B) and interactive filtering (I), gathered by the target language (multilingual runs appear in several groups, in order to present the individual scores on each target language: in this case, the name of the run has been suffixed with the target language). Best result is obtained on monolingual English, but for the only participant that tried multilingual runs, the results obtained for the different target languages (English and French) are comparable (the results for crosslingual are 90% of the results for monolingual runs of this participant).

Table 4. Scores for batch and interactive filtering runs, gathered by target language, sorted by F-score

	team	run	langs	n_rel	n_rel_ret	prec	recall	F-score	utility	anticip
B	IMAG	IMAG_1	eng-eng	1597	413	0.26	0.30	0.21	0.21	0.43
B	UAIC	uaic_4.eng	eng-eng	1597	1267	0.09	0.66	0.13	0.05	0.73
B	UAIC	uaic_1	eng-eng	1597	1331	0.06	0.69	0.09	0.03	0.75
B	UAIC	uaic_2.eng	eng-eng	1597	1331	0.06	0.69	0.09	0.03	0.75
B	UAIC	uaic_3.eng	eng-eng	1597	1507	0.06	0.82	0.09	0.03	0.86
B	IMAG	IMAG_2	eng-eng	1597	109	0.13	0.09	0.07	0.16	0.22
B	IMAG	IMAG_3	eng-eng	1597	66	0.16	0.06	0.07	0.22	0.14
B	SINAI	topics_1	eng-eng	1597	940	0.02	0.50	0.04	0.00	0.57
B	SINAI	googlenews_2	eng-eng	1597	196	0.01	0.08	0.01	0.13	0.10
B	UAIC	uaic_4.fre	eng-fre	2421	1120	0.09	0.44	0.12	0.05	0.58
B	UAIC	uaic_3.fre	eng-fre	2421	1905	0.06	0.75	0.10	0.03	0.83
B	UAIC	uaic_2.fre	eng-fre	2421	1614	0.06	0.67	0.09	0.02	0.76
B	UAIC	uaic_4	eng-eng/fre	4018	2387	0.07	0.56	0.11	0.02	0.72
B	UAIC	uaic_3	eng-eng/fre	4018	3412	0.05	0.81	0.08	0.02	0.85
B	UAIC	uaic_2	eng-eng/fre	4018	2945	0.05	0.70	0.07	0.02	0.80

The scores for adaptive filtering in the interactive task are worse than the scores obtained on the batch filtering, but the language pairs and the participants are not the same[11] which makes the comparison difficult. We also note than both batch and adaptive results for the INFILE 2009 campaign are worse than the results obtained for the adaptive task in the INFILE 2008 edition, although the same document collection

10 Following values are presented: number of relevant documents in collection (n_rel), number of relevant documents retrieved (n_rel_ret), precision (prec), recall, F-score, utility value and anticipation (anticip).

11 The UOWD run is monolingual English as most of batch filtering runs, but the submitted run has been obtained on only a subset of the document collection due to technical problem of the participant, which explains the poor scores.

and topics were used: the only team that participated to the track both years gives a more detailed comparison on the differences for their runs in his report [11].

To measure the impact of the use of the simulated feedback in the interactive task, we present in Fig. 4 the evolution of the scores across time, for batch filtering runs and adaptive filtering runs. Again, the language pair being different for batch runs and adaptive runs, the comparison is not easy: we choose the compare the results according to the target language (the language of the documents).

Fig. 3. Examples of evolution of the scores across time for batch and interactive tasks

These results are not conclusive about the impact of the use of simulated feedback: there is no obvious improvement of the behavior of the filtering systems across time when using adaptive techniques. One of the reasons may be that the feedback was (voluntarily) limited in the interactive task: the participants considered that the number of authorized feedbacks was too small to have efficient learning. Further experiments with more feedbacks allowed should be conducted.

Results for originality measures for both batch and interactive tasks are presented in Table 5 and 6, gathered by target language. Table 5 presents the originality scores for every run that has the same target language (i.e. the number of relevant documents

that this particular run uniquely retrieves). Since this global comparison may not be fair for participants who submitted several runs, which are presumably variants of the same technique and will share most of the relevant retrieved documents, we present in Table 6 the originality scores using only one run for each participant (we chose the run with the best recall score). We see here that participants with lower F-scores can have a better originality score: even if their precision is not as good, they manage to retrieve documents not retrieved by the other participants. However, due to the small number of participants, the relevance of the originality score is arguable in this context, since it seems to be correlated to the recall score.

Table 5. Originality scores on all runs, gathered by target language

originality on all runs					
target lang=eng			**target lang=fre**		
team	**run**	**originality**	**team**	**run**	**originality**
UAIC	uaic_3	39	HossurTech	hossur-tech-004	177
HossurTech	hossur-tech-001	18	UAIC	uaic_3	82
SINAI	googlenews_2	15	UAIC	uaic_2	0

Table 6. Originality scores on best run by participant (besrt recall score), gathered by target language

UAIC	uaic_4	4
IMAG	IMAG_1	1
UAIC	uaic_1	0
IMAG	IMAG_3	0
UOWD	base	0
UAIC	uaic_2	0
IMAG	IMAG_2	0

6 Conclusion

The INFILE campaign has been organized in 2009 for the second time in CLEF, to evaluate adaptive filtering systems in a cross-language environment. The document and topic collection were the same as the 2008 edition of the INFILE@CLEF track. Two tasks have been proposed: a batch filtering task and an adaptive filtering task, that used an original setup to simulate the incoming of newswires documents, and the interaction of a user through a simulated feedback. We had in 2009 more participants than in INFILE previous edition and more results to analyze. Best result are still obtained on monolingual runs (English), but the difference is not important from crosslingual runs. However, the innovative crosslingual aspect of the task has still not

been fully explored, since most runs were monolingual English and no participant used the Arabic topics or documents. Also, the lack of participation for the adaptive task does not provide enough data to compare batch techniques to adaptive techniques and does not allow to conclude on the interest of the use of a feedback on the retrieved documents to improve filtering techniques.

References

1. Besancon, R., Chaudiron, S., Mostefa, D., Hamon, O., Timimi, I., Choukri, K.: Overview of CLEF 2008, INFILE Pilot Track (2008)
2. Robertson, S., Soboroff, I.: The trec 2002 filtering track report. In: Proceedings of The Eleventh Text Retrieval Conference (TREC 2002). NIST (2002)
3. Fiscus, J., Wheatley, B.: Overview of the tdt 2004 evaluation and results. In: TDT 2002. NIST (2004)
4. Yang, Y., Yoo, S., Zhang, J., Kisiel, B.: Robustness of adaptive filtering methods in a cross-benchmark evaluation. In: Proceedings of the 28th Annual international ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, pp. 98–105 (2005)
5. Soboroff, I., Robertson, S.: Building a filtering test collection for TREC 2002. In: Proceedings of The Eleventh Text Retrieval Conference (TREC 2002). NIST (2002)
6. Thomson, P.: Description of the PRC CEO algorithm for TREC-2. In: TREC-2: Text retrieval conference No2. NIST special publications, Gaithersburg (1994)
7. Van Rijsbergen, C.: Information Retrieval. Butterworths, London (1979)
8. Hull, D., Roberston, S.: The trec-8 filtering track final report. In: Proceedings of the Eighth Text REtrieval Conference (TREC-8). NIST (1999)
9. The topic detection and tracking phase 2 (TDT2) evaluation plan. NIST (1998), `http://www.nist.gov/speech/tests/tdt/1998/doc/tdt2.eval.plan.98.v3.7.pdf`
10. Voorhees, E.: The trec-8 question answering track report. In: Proceedings of the Eighth Text REtrieval Conference (TREC-8). NIST (1999)
11. Qamar, A.M., Gaussier, E., Denos, E.: Batch Document Filtering Using Nearest Neighbor Algorithm. In: Working Notes of CLEF 2009 (2009)

Batch Document Filtering Using Nearest Neighbor Algorithm

Ali Mustafa Qamar[1,2], Eric Gaussier[1,2], and Nathalie Denos[1,3]

[1] Laboratoire d'Informatique de Grenoble (LIG)
[2] Université Joseph Fourier
[3] Université Pierre Mendès France
{ali-mustafa.qamar,eric.gaussier,nathalie.denos}@imag.fr

Abstract. We propose in this paper a batch algorithm to learn category specific thresholds in a multiclass environment where a document can belong to more than one class. The algorithm uses the k-nearest neighbor algorithm for filtering the 100,000 documents into 50 profiles. The experiments were run on the English corpus. Our experiments gave us a macro precision of 0.256 while the macro recall was 0.295. We had participated in the online task in INFILE 2008 where we had used an online algorithm using the feedbacks from the server. In comparison with INFILE 2008, the macro recall is significantly better in 2009, 0.295 vs 0.260. However the macro precision in 2008 were 0.306. Furthermore, the anticipation in 2009 was 0.43 as compared with 0.307 in 2008. We have also provided a detailed comparison between the batch and online algorithms.

1 Introduction

The INFILE (INformation FILtering Evaluation) [2] track is a cross-language adaptive filtering evaluation campaign, a part of the CLEF (Cross Language Evaluation Forum) campaign. It is composed of 100,000 Agence France Press (AFP) comparable newswires covering the years 2004 to 2006, and written in either Arabic, English or French. News articles in different languages are not necessarily translation of each other, and are given in XML format. The goal of the INFILE campaign is to filter these 100,000 documents into 50 topics (plus a category 'other'). Out of 50 topics, 30 are related to general news and events (e.g. national and international affairs, sports, politics etc.), whereas the rest concerns scientific and technical subjects. A document can belong to zero, one or more topics, each topic being described by a set of sentences. The topics or profiles have been created by competitive intelligence professionals. It extends the TREC 2002 filtering track. In comparison with INFILE 2008, where there was only an online task, an additional batch filtering task was added in 2009. As opposed to the online task, where the server provides the documents one by one to the user, all of the documents are provided beforehand in the batch task. This explains the fact that feedback is not possible in the batch task. We had

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 354–361, 2010.

participated in the online task in 2008 [3], and restricted ourselves to the batch one in 2009.

The k-nearest neighbor (kNN) algorithm is a supervised learning algorithm, largely investigated due to its simplicity and performance. It aims at finding the k nearest neighbors of an example x (based either on similarity or distance) and then finding the most represented class in the nearest neighbors. Previous studies have shown that similarity measures are more appropriate than distance ones when dealing with texts (see e.g. [5]). We thus rely in this work on the cosine measure rather than the Euclidean or Mahanalobis distances.

In this paper, we develop a batch algorithm to learn category-specific thresholds in a multiclass environment. Our algorithm uses the kNN algorithm along with the cosine similarity, in order to filter documents into various topics. The rest of the paper is organized as follows: Section 2 describes the batch algorithm developed for the INFILE campaign followed by its comparison with Online algorithm of 2008 in Section 3. The experiments and results are discussed in Section 4 while we conclude in Section 5.

2 Batch Algorithm for the INFILE Campaign

In order to filter the documents into various topics, we use a similarity measure between new documents and topics, along with a set of thresholds on this similarity that evolves over time. The similarity between a new document d, to be filtered, and a topic t_i can be given as:

$$\mathrm{sim}(t_i, d) = \alpha * \underbrace{cos(t_i, d)}_{s_1(t_i,d)} + (1-\alpha)\underbrace{\max_{(d' \neq d, d' \in t_i)} cos(d, d')}_{s_2(t_i,d)} \quad (1)$$

where $\alpha \in$ [0,1]. The similarity given in equation 1 is based on two similarities: one based on a direct similarity between the new document and the topic (given by $s_1(t_i, d)$), and another one between the new document and the set of documents already assigned to the topic ($s_2(t_i, d)$). One might think that only the first similarity would suffice. However, this is not the case since the topics and the documents do not share the same kind of structure and content. The second similarity helps us to find documents which are closer to documents which had already been assigned to a topic. α is used to control the importance of the two similarities. In the beginning, when no documents are assigned to any topic, only the similarity between a topic and the new document, $s_1(t_i, d)$, is taken into account. This similarity is used to find a certain number of nearest neighbors for each of the document (10 in our case) which eventually enables us to use the second similarity. A threshold was used for each of the 50 topics. We now describe the batch algorithm to filter the documents into various profiles/topics. As already mentioned, the feedback is not possible in this case since the complete set of documents is transferred to the user in one go.

Batch Algorithm

Construction of initial set:
for each topic i $(i \in \{101,102,...,150\}$
find 10 nearest neighbors based on $s_1 = cos(t_i, d)$
for each nearest neighbor d found
$t_i \Leftarrow d$
Assignment of remaining documents to topics:
$\alpha = 0.7$
for each topic i
$\theta_i = \min_{d \in t_i} \text{sim}(t_i, d)$
for each document d
for each topic i
if $(\text{sim}(t_i, d) \geq \theta_i)$
$t_i \Leftarrow d$
$\theta_i = \min(\theta_i, \min_{d \in t_i} \text{sim}(t_i, d))$

Yang et al. [7] have described a similar method, whereby they learn category-specific thresholds based on a validation set. An example is assigned to a particular category only if its similarity with the category surpasses a certain learned threshold. In contrary, we do not have a validation set to learn thresholds, however, we create a simulated one, by finding nearest neighbors for each of the 50 topics.

3 Comparison with Online Campaign 08

We present here, a detailed comparison between the batch algorithm we used in 2009 and the online algorithms we developed for the online campaign in 2008. We present here the two algorithms developed, a general algorithm which makes use of the feedbacks to build an initial set of documents and its simplication which does not use feedbacks.

Online Algorithm (General)

$\alpha = 0.7$, $\theta^1 = 0.42$
for each new document d
for each topic i
% $(i \in \{101,102,...,150\}$
Construction of initial set:
if ($l_i < 10$)
if $(s_1(t_i, d) > \theta^1)$
Ask for feedback (if possible) and $t_i \Leftarrow d$ if feedback positive

Assignment of remaining documents to topics:
else if $(\text{sim}(t_i, d) > \theta_i^2)$
$t_i \Leftarrow d$
where $\theta_i^2 = \min_{d \in t_i} \text{sim}(t_i, d)$

---

where l_i represents the number of documents assigned to a topic i. For each topic, two thresholds are used: the first one (θ^1) allows filtering the documents in the early stages of the process (when only a few documents have been assigned to the topic). The value chosen for this threshold was 0.42. The second threshold (θ_i^2), however, works with the global similarity, after a certain number of documents have been assigned to the topic.

The main difference between the two algorithms (batch and online) lies in the manner in which we construct the initial set of documents relevant to the topics. In the batch algorithm, we just rely on finding the 10 nearest neighbors for each topic, with the assumption that the nearest neighbors for a topic would, in general, belong to the topic under consideration. However, for the online algorithm, we use feedbacks (limited to 50) to add a document to a profile if the similarity between a topic t_i and a document d is greater than a certain threshold (θ^1). We repeat this procedure until either 10 documents have been added to each of the 50 topics or we have seen all of the 100,000 documents. Hence it is possible that a certain topic has less than 10 documents after the construction of the initial set. On the contrary, the use of nearest neighbors in the batch algorithm ensures that each topic has exactly 10 documents after the buildup of the initial set.

Furthermore, as the online algorithm builds the initial set of documents based on the threshold θ^1, hence, it is very important that this threshold is chosen very carefully (a dry run was used to tune the value of θ^1 during the online campaign in 2008). On the other hand, the batch algorithm does not use any threshold during the construction of the initial set.

The second phase of the two algorithms, where we assign the remaining documents to topics, is similar except the fact that we update the threshold θ_i in the batch algorithm, only if the current threshold is smaller than the previously stored one. However, the online algorithm does not make use of previously stored value of the threshold θ_i^2. This means that the batch algorithm is more lenient in assigning new documents to topics as compared to the online algorithm.

In addition to the general version, a simplified version of the online algorithm was also developed for INFILE 2008. This algorithm neither uses any feedback nor builds an initial set of documents. It does not update the threshold θ_i^2 unlike the general algorithm. Here, a threshold θ is derived from θ^1 and θ^2, according to equation 1, which integrates the two similarities θ^1 and θ^2 operate upon:

$$\theta = \alpha * \theta^1 + (1 - \alpha) * \theta^2 \tag{2}$$

The threshold θ replaces θ_i^2 of the online algorithm. As this simplified algorithm does not build an initial set of documents, hence it cannot use $s_2(t_i, d)$ unless some document has been assigned to the topic t_i.

4 Experiments

We have run our algorithm on the INFILE English corpus. For all of the documents, stemming was performed using Porter's algorithm [6]. This was followed by the removal of stop-words, XML tags skipping and the building of a document vector (which associates each term with its frequency) using the Rainbow package [4]. A single run was submitted during the INFILE campaign. Initially, 10 nearest neighbors were found for each of the document based on the similarity s_1 (between a document and the topic). These documents were subsequently used to compute s_2. The experiment was divided into 4 sub-parts, each sub-part being run in parallel to increase the efficiency. However, this setting meant that the thresholds for the 50 topics were different for the different sub-parts.

There are 1597 documents relevant to one or more topics in the INFILE data. We compare the batch algorithm of 2009 with the general online algorithm and its simplified version developed by us in 2008. It may be recalled that for Run 3 (run2G), θ^1 was chosen to be 0.45 while θ^2 was set to 0.8. Similarly for Run 4, the values for θ^1 and θ^2 were 0.4 and 0.7 respectively.

The results for the different runs were evaluated based on different measures, namely, precision, recall, F-measure, linear utility, anticipation (added in 2009) and detection cost (see [1] and [2]). Utility is based on two parameters: importance given to a relevant document retrieved and the cost of a non-relevant document retrieved. Anticipation measure is designed to give more importance to systems that can find the first document in a given profile.

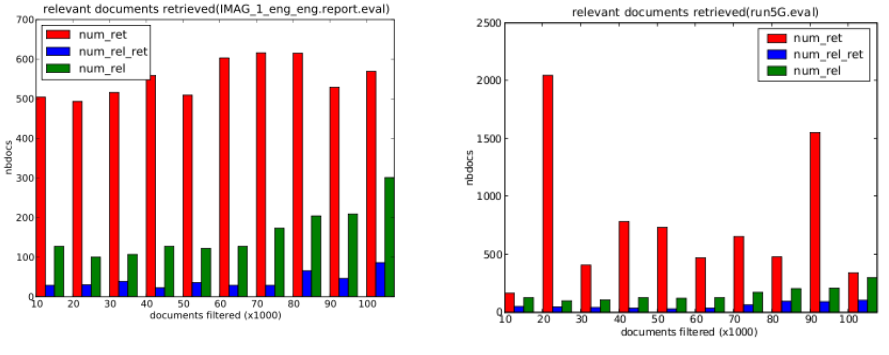


Fig. 1. Number of documents retrieved for Run 1(left) and 2 (right)

Figure 1 and 2 give us an insight on the number of relevant documents retrieved during the different runs. From these two figures, we do not see a significant change for Run 1 and Run 3, in terms of the number of documents retrieved during the entire process. However, Run 2 returns much more documents between 10,000-20,000 and 80,000-90,000 documents. Similarly Run 4 retrieves more documents between 10,000-40,000 and 50,000-70,000 documents.

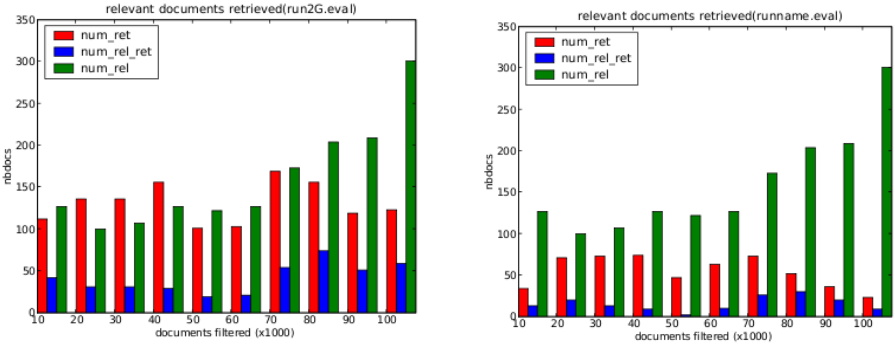


Fig. 2. Number of documents retrieved for Run 3 (left) and 4 (right)

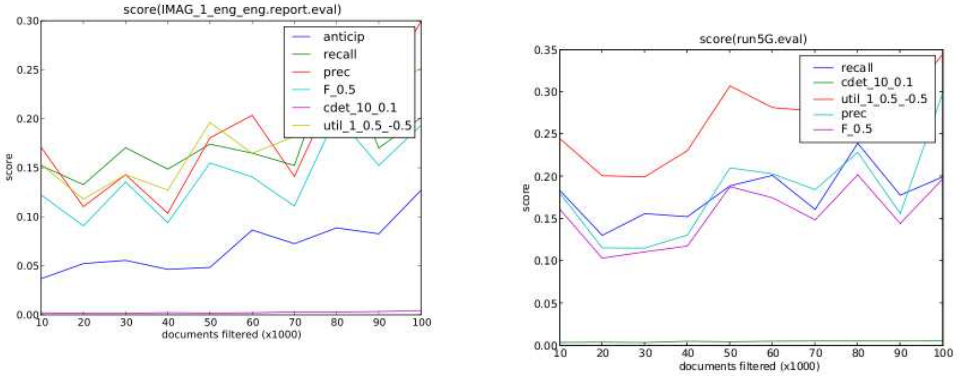


Fig. 3. Score Evolution for Run 1 (left) and 2 (right)

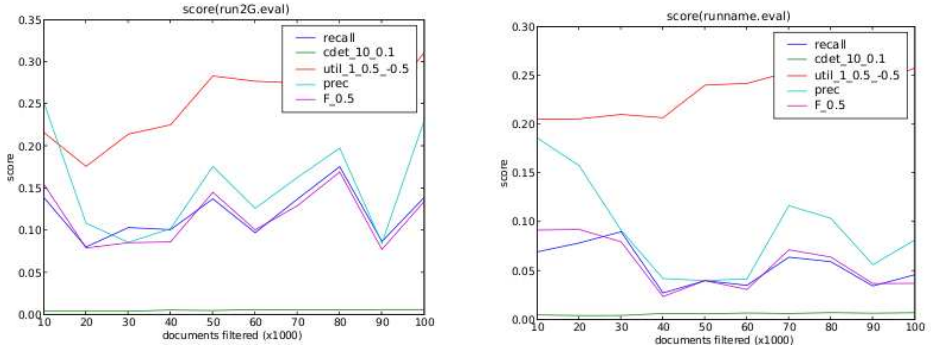


Fig. 4. Score Evolution for Run 3 (left) and 4 (right)

The evolution of these measures, computed at different times in the process, after each 10,000 documents, are given in Figures 3 and 4. The curve, at the bottom represents the detection cost. Similarly, for Run 1, the curve just above the one meant for detection cost, describes anticipation. For Run 1, all of the measures randomly vary but increase significantly as compared to the initial values (for example, 0.04 in the beginning vs 0.125 at the end for anticipation, 0.12 to 0.19 for the F-measure etc.) during the course of the filtering process. For Run 2, all of the measures, except utility and precision (0.18 vs 0.30), randomly vary but remain the same at the end. All measures vary for Run 3 and 4 during the filtering process. However only the values of linear utility increase and the final values are higher than the initial values.

Table 1. Detail about the different runs

Name	Campaign	Algorithm	Doc. ret	Doc. ret - relevant
Run 1 IMAG_1	Batch 09	Batch (w/o feedback)	5513	413
Run 2 run5G	Online 08	Online (with feedback)	7638	601
Run 3 run2G	Online 08	Online (w/o feedback)	1311	411
Run 4 runname	Online 08	Online (w/o feedback)	546	152

Table 1 describes the different runs along with the number of documents retrieved and the number of relevant documents found. We can compute various measures like micro precision, micro recall etc. from this table. Run 3 has the highest micro precision whereas Run 2 has got the highest micro recall. These values are computed on the entire corpus.

Table 2. Run Scores

	Macro_P	Macro_R	Macro_F	Macro_LU	Macro_DC	Anticipation
Run 1	0.256	0.295	0.206	0.205	0.002	0.430
Run 2	0.306	0.260	0.209	0.351	0.007	0.307
Run 3	0.357	0.165	0.165	0.335	0.008	0.317
Run 4	0.366	0.068	0.086	0.311	0.009	0.207

Table 2 describes the macro values for the different runs. These values represent the average score over the complete set of 50 profiles. P represents precision, R represents recall, F represents F-measure, LU represents linear utiliy while DC represents detection cost. Run 1 has the best macro recall (0.295) as compared with all the runs. The macro F-measure for the Run 1 and Run 2 are significantly greater than that of Run 3 and 4. However, Run 2 surpasses Run 1 in terms of macro precision. The overall macro detection cost is very low in all of these runs, with Run 1, being the most economical. This is a strong point for

these algorithms. The macro linear utility of Run 2 is greater than that of Run 1. On contrary, anticipation for Run 1 is significantly better than that of Run 2.

We can easily conclude from these results, that the use of limited number of feedbacks (only 50 i.e. one per topic) did not help to get very good results, although it helped to increased the micro recall.

5 Conclusion

We have presented, in this paper, a simple extension of the kNN algorithm using thresholds to define a batch filtering algorithm. The results obtained can be deemed encouraging as the macro F-measure equals approximately 20%, for a collection of 100,000 documents and 50 topics, out of which only 1597 documents are relevant. In comparison with online results of 2008, we have a much better macro recall (almost 30% against 26% in 2008) along with a lower macro detection cost (0.002 vs 0.007) and a much better anticipation (0.430 vs 0.307). Considering the evolution of different measures, we had observed that the values for all of the measures increase, with the increase in the number of documents filtered. The main difference between the batch and online algorithms lies in the way the initial set of documents is constructed. In batch algorithm, the initial set is built from finding the 10 nearest neighbors for each of the profile. Whereas feedbacks are used in the online algorithm to construct the initial set of documents. We can also conclude from the results that, the use of a limited number of feedbacks does not help to get very good results.

References

1. Besancon, R., Chaudiron, S., Mostefa, D., Hamon, O., Timimi, I., Choukri, K.: Overview of CLEF 2008 infile pilot track. In: Working Notes of the Cross Language Evaluation Forum, Aarhus, Denmark, pp. 17–19 (September 2008)
2. Besancon, R., Chaudiron, S., Mostefa, D., Timimi, I., Choukri, K.: The infile project: a crosslingual filtering systems evaluation campaign. In: ELRA (ed.), Proceedings of LREC 2008, Morocco (May 2008)
3. Bodinier, V., Qamar, A.M., Gaussier, E.: Working notes for the infile campaign: Online document filtering using 1 nearest neighbor. In: Working Notes of the Cross Language Evaluation Forum, Aarhus, Denmark, September 17-19 (2008)
4. McCallum, A.K.: Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering (1996)
5. Qamar, A.M., Gaussier, É., Chevallet, J.-P., Lim, J.-H.: Similarity learning for nearest neighbor classification. In: ICDM 2008, pp. 983–988 (2008)
6. Jones, K.S., Willett, P. (eds.): Readings in information retrieval. Morgan Kaufmann Publishers Inc., San Francisco (1997)
7. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: SIGIR 1999, USA, pp. 42–49. ACM Press, New York (1999)

UAIC: Participation in INFILE@CLEF Task

Cristian-Alexandru Drăguşanu, Alecsandru Grigoriu, and Adrian Iftene

UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
{cristian.dragusanu,alecsandru.grigoriu,adiftene}@info.uaic.ro

Abstract. This year marked UAIC's first participation at the INFILE@CLEF competition. The purpose of this campaign is the evaluation of cross-language filtering systems, which is to successfully build an automated system that separates relevant from non-relevant documents written in different languages with respect to a given profile. For the batch filtering task, participants are provided with the whole document collection and must return the list of relevant documents for each topic. We achieved good results in filtering documents, also obtaining the highest originality score, when having English as target language. Our team was also the only one who submitted runs for cross-lingual and multilingual batch filtering, with French and English/French as target languages. A brief description of our system, including presentation of the Parsing, Indexing and Filtering modules is given in this paper, as well as the results of the submitted runs.

1 Introduction

INFILE@CLEF[1] (information filtering evaluation) extends the TREC 2002 filtering track. In comparison, it uses a corpus of 100,000 Agence France Press comparable newswires for Arabic, English and French [1]. The participants received news collections containing 100,000 news articles for each language (English, French and Arabic), stored in directories, each news article being in a separate file. The articles in the different languages are not translations of one another, they are independent articles. Also, the participants received 50 topics for all three languages.

In the batch filtering task, competitors must compare each topic in a source language to the documents in the target languages. Every source/target languages are allowed: results can be provided for monolingual filtering, cross lingual filtering or multilingual filtering (with a mixed set of documents from different target languages), as long as they use only the topics in the source language (provided translations of the topics should not be used for cross lingual filtering, either directly or for training).

2 UAIC System

Our system has three main modules: first module responsible for XML parsing, second module which indexes the files, and the third module that does the filtering.

[1] INFILE@CLEF: http://www.infile.org/

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 362–365, 2010.

Module for XML Parsing: First of all, we parse the XML files in order to extract relevant content from the documents (which are in News-ML format). The *Indexing module* needs several data from the NewsML documents, essential for the Filtering run. From the given files we need to focus on *DateID*, *NewsItemId*, *Slugline*, *Headline*, *DataContent*, *City*, *Country* and *Filename* (aldfgl important to the indexing part). The Topics store a minimum amount of data therefore we will focus only on: *Topic number, Title, Description, Narration, Keywords* and *Sample* (used later on by the Filtering module).

Indexing Module: For indexing we use Lucene [3], a suite of free libraries used both for indexing and for searching. All documents are parsed by XML Parsing, one by one, and the representative fields are sent to the Indexing module as parameters.

Filtering Module: The Filtering part can be viewed as a separate application, even though the used modules are the same as in the Indexing part. In the Filtering part, the file containing the 50 topics (in XML format) is parsed by XML Parsing module. Then, for each one of the 50 topics, a number of fields are extracted and sent to the Filtering module. The Filtering module receives the topic details, sorts and filters individual words from all fields and generates a search query based on the most frequent and relevant words from the topic. The search query is designed to optimize the index search by adding specific terms to be searched in specific index fields (for example *Slugline*, *Headline*, etc.) and by adding different importance to each field. When the query string is fully generated, it's passed as a parameter on to the Indexing module, which will return a list of documents matching the query.

The topic is parsed by the Parsing module, splitting it in multiple fields. All prepositions are removed from the fields, so they won't be used in the search algorithm, being too general to return any relevant results. The most frequent 5 words and the negated expressions are extracted from all fields. Because the News-ML format can contain information about the date or location of the article, a heuristic algorithm was implemented to extract such information from the fields, so the search results can be refined.

The extracted items are combined in different ways, so a general search query can be formed and used to search the previously created index for matching documents. The most frequent 3 words in the topic title are marked as the most important words in the query and are mandatory in all search fields (they are marked with a leading "+").

The words that will be searched in the *Headline* field include the most frequent word from each of the *Description*, *Narration*, *Sample* and *Keywords* topic fields. The *Slugline* query also contains the most frequent word from each of the same topic fields, except that all 5 frequent words from *Keywords* are used. Extracted dates and locations are added to the query string, to be searched in the *DateId*, *City* and *Country* fields. The rest of the frequent words are searched in the DataContent field.

The results will be written in the output file and the next topic is processed. If there are no more topics to be processed, the application stops.

3 Submitted Runs

For our runs the search was made in 2 languages, English and French, using topics in English. Each language archive contains 100,000 news articles, stored in directories according to the following organization: `<language>/<year>/<month>/<day>`.

There are 50 topics for each language, but only the English topics were used by us for testing. The 50 topics are stored in one XML file, encoded in UTF-8. Usually, on a search on all fields, using the most optimized algorithm, the matching documents are between 0.5-2 % from the total number of documents.

Filtering based on topic 101 (first topic from the English set), the final results are: 178 hits in English (2004), 102 hits in English (2005), 92 hits in English (2006), 295 hits in French (2004), 167 hits in French (2005) and 188 hits in French (2006). So, from a total of 372 hits for English and 650 hits for French, there are 1022 hits in total out of 200.000 documents, so the percent represented by number of hits is 0,511%.

4 Results

CLEF 2009 INFILE track had two types of tasks: batch filtering task (3 participants) and interactive filtering task (2 participants). We participated at the batch filtering task, with 4 runs. For the batch filtering task, there were 9 runs on English and 3 runs on French. As noticed, all participants chose English as the source language, and no participants used Arabic as source, nor as target language.

The best overall results for monolingual English, based on the *F_score* value, were obtained, in order, by: IMAG (France), UAIC (Romania) and SINAI (Spain). The most interesting approach was used by IMAG [2] and was based on finding the k-nearest neighbor, along with cosine similarity.

We submitted 4 runs, one run Eng/Eng (with English as source language and target language) and three runs Eng/Eng-Fre (with English as source language and with English and French as target languages). Details are presented in Table 1.

Table 1. UAIC Runs 3 and 4

Run	Lang.	Number of Documents			prec	recall	F_score	Antic
		Ret.	Rel.	Rel. ret.				
Run 3	Eng/Eng	75.915	1.597	1.507	0.06	0.82	0.09	0.86
	Eng/Fre	67.124	2.421	1.905	0.06	0.75	0.10	0.83
	Eng/Eng-Fre	143.039	4.018	3.412	0.05	0.81	0.08	0.85
Run 4	Eng/Eng	33.793	1.597	1.267	0.09	0.66	0.13	0.73
	Eng/Fre	21.591	2.421	1.120	0.09	0.44	0.12	0.58
	Eng/Eng-Fre	55.384	4.018	2.387	0.07	0.56	0.11	0.72

Based on the official Overview of CLEF 2009 INFILE track, our best results were achieved on Run 3 (the highest recall value) and Run 4 (highest precision from our runs). Although the precision on Run 4 was significantly lower than the precision achieved by other teams, our high recall value brought us on second place overall, based on the *F_score* value (0.13). Besides the monolingual English comparison, we

also submitted three runs for cross-lingual English/French and multilingual English/English-French. However, because we were the only team which submitted such runs, we cannot correctly assess our results. Our team also achieved very good results on both experimental measures introduced this year: originality and anticipation. Run 3 was the submission which got the highest originality score on both categories (All runs and Best run), with English as target language. This run also obtained a very high originality score with French as target language. It also achieved the highest anticipation from all submissions made by all the teams, with almost 30% advantage from the nearest score of the rest of the teams.

Table 2. Originality of Run 3

Source/Target	Originality on all runs	Originality on best run
Eng/Eng	39	267
Eng/Fre	82	1292

Our main issue was the very low precision, due to a high number of irrelevant documents retrieved. This could be improved by adjusting the search query string and also by filtering the retrieved documents based on the Lucene score. This way, the system should obtain more relevant results, therefore improving the precision value.

5 Conclusions

This paper presents the UAIC system which took part in the INFILE@CLEF 2009 competition. This year is the second edition of the INFILE campaign and our first participation on this project. We designed a system formed of several modules: the *parsing module*, which retrieves the relevant content from the documents, the *indexing module* done with Lucene and the *filtering module* which generates a Lucene query and extract from Lucene index the relevant documents. From a total of 4 submitted runs, the best results were obtained on Run 3, when English was used as both the source and the target language. Run 3 achieved the second-best *F_score* from all teams (having the highest recall value) and also the highest originality score from all submitted runs. We were also the only ones who submitted runs for cross-lingual and multilingual filtering, with French and English-French languages as targets.

References

1. Besançon, R., Chaudiron, S., Mostefa, D., Hamon, O., Timimi, I., Choukri, K.: Overview of the CLEF 2008 INFILE Pilot Track. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, Springer, Heidelberg (2009)
2. Besançon, R., Chaudiron, S., Mostefa, D., Timimi, I., Choukri, K., Laïb, M.: Overview of CLEF 2009 INFILE track. Track. In: Working Notes of the Cross Language Evaluation Forum (CLEF 2009), Corfu, Greece (September 2009)
3. Hatcher, E., Gospodnetic, O.: Lucene in action. Manning Publications Co. (2005)

Multilingual Information Filtering by Human Plausible Reasoning

Asma Damankesh[1], Farhad Oroumchian[1], and Khaled Shaalan[2]

[1] University of Wollongong in Dubai, P.O. Box 20183, Dubai, UAE
[2] The British University in Dubai , P.O. Box 502216, Dubai, UAE
adamankesh@acm.org,
FarhadOroumchian@uowdubai.ac.ae,
khaled.shaalan@buid.ac.ae

Abstract. The theory of Human Plausible Reasoning (HPR) is an attempt by Collins and Michalski to explain how people answer questions when they are uncertain. The theory consists of a set of patterns and a set of inferences which could be applied on those patterns. This paper, investigates the application of HPR theory to the domain of cross language filtering. Our approach combines Natural Language Processing with HPR. The documents and topics are partially represented by automatically extracted concepts, logical terms and logical statements in a language neutral knowledge base. Reasoning provides the evidence of relevance. We have conducted hundreds of experiments especially with the depth of the reasoning, evidence combination and topic selection methods. The results show that HPR contributes to the overall performance by introducing new terms for topics. Also the number of inference paths from a document to a topic is an indication of its relevance.

1 Introduction

Human Plausible Reasoning (HPR) is a relatively new theory that tries to explain how people can draw conclusions in an uncertain and incomplete situation by using indirect implications. For 15 years, Collins and his colleagues have been investigating the patterns used by human to reason under uncertainty and incomplete knowledge [1]. The theory assumes that a large part of human knowledge is represented in "dynamic hierarchies" that are always being modified, or expanded. This theory offers a set of frequently recurring inference patterns used by people and a set of transformations on those patterns [1]. A transformation is applied on an inference pattern based on a relationship (i.e. generalization and specialization) to relate available knowledge to the input query. Elements of expression in the core theory have been summarized in Fig. 1. The theory has many parameters for handling uncertainty but it does not explain how these parameters could be calculated which is left for implementations and adaptations. Interested readers are referred to references [1], [2] and [3]. Different experimental implementation of the theory such as adaptive filtering [4], XML retrieval [5] or expert finding [6] have proved the flexibility and usefulness of HPR in

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 366–373, 2010.

the Information Retrieval (IR) domain. All the works on HPR shows that it is a promising theory which needs more investigation to be applicable. This research is about creating a framework for multilingual filtering and information retrieval where all aspects of retrieval in this environment are represented as different inferences based on HPR. In this framework, documents and topics are partially represented as a set of concepts, logical terms and logical statements. The relationships of concepts are stored in a knowledge base regardless of their language of origin. Therefore, by inference, we can retrieve relevant documents or topics of any language stored in the knowledge base. This paper is structured as follows. Sections 2 describes the system architecture. Section 3 explains the experimental configurations. Section 4 summarizes the results and section 5 is the conclusion.

Table 1. Elements of expression in The Core Plausible Reasoning Theory

Baghdad is the capital of Iraq	
Referent $r_1, r_2, ..$ or $r_1, r_2, ...$	e.g. Baghdad
Argument $a_1, a_2, ..$ or $F(a)$	e.g. Iraq
Descriptor $d_1, d_2, ..$	*e.g.* Capital
Term $d_1(a_1), d_1(a_2), d_2(a_3), ..$	e.g. Capital(Iraq)
Statement $d_1(a_1) = r_1, d_1(a_2) = r_1, r_2, .., d_2(a_3) = r_3, ..$	*e.g.* $Capital(Iraq) = Baghdad$
Dependency between terms : $d_1(a_1) \leftrightarrow d_2(a_2)$	
e.g. latitude(place) $\leftrightarrow$ average-temp(place): Moderate, Moderate,Certain	
(translation): i am certain that latitude constrains average temperature with moderate reliability and that the average temperature of the place constraints the latitude with moderate reliability	
Implication between statements : $d_1(a_1) = r_1 \leftrightarrow d_2(a_2) = r_2$	
e.g. grain(place)=rice,... $\leftrightarrow$ rainfall(place)=heavy: high, Low,Certain	
(translation): i am certain that if a place produce rice, it implies that the place has heavy rainfall with high reliability, but if a place has heavy rainfall it only implies that the place produces rice with low reliability	

2 System Architecture

System architecture is depicted in Fig. 1. The Text Processor unit processes a document into a set of concepts, logical terms and logical statements. Document Representation unit, assigns a weight to each term. Topic Retrieval unit finds topics that have been indexed by the given terms and computes a certainty value. Inference Engine applies transforms of Human Plausible Reasoning to the terms in the document representation that exist in the knowledge base and generates a set of new terms. These new terms are used to expand the document representation. Then the new terms are given to Topic Retrieval to retrieve matching

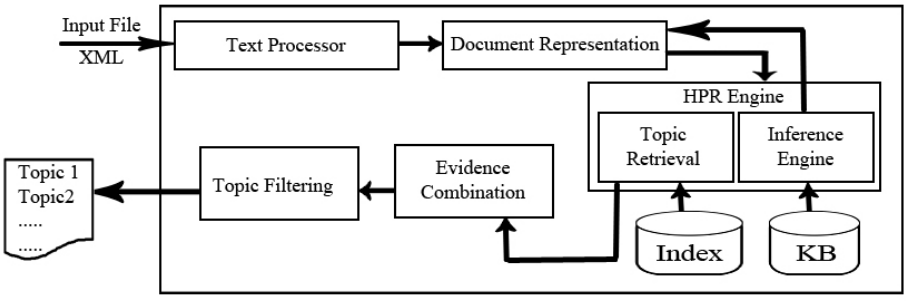


Fig. 1. Topic Retrieval and Filtering Unit

topics. The process of expanding the document representation and retrieving topics will be repeated several times. Each document term could be processed more than once (different inferences can reach the same term via different paths). A document could be linked to the same topic through multiple inferences and paths. Therefore, multiple certainty values could be assigned to a topic through different reasoning paths and terms. Topic Filtering unit is responsible for combining these certainty values into a single certainty value that represents the confidence in how well a topic can be inferred from a document representation. the KB is created by the Information Extractor Unit depicted in Fig. 2. This unit takes a list of file names and one by one reads through these files. Each file contains a document. The document goes through a pre-processing for normalizing the text. Then Part of Speech Tagging and stemming are applied. For POS tagging we have used Monty Tagger [7] and for stemming we have used a Python version of Porter stemmer [8]. The Text Miner is a rule based program that takes in the part of speech tagged text and extracts the relationships among the concepts. At the moment these rules are based on the clue words in the text. For example, they use propositions to infer relationships between two concepts around the proposition. To build the Knowledge Base, the Build KB unit takes in these relationships and calculates a confidence value based on the frequencies of occurrences of relationships.

The KB normally contains KO (Kind of), ISA (is a) and PO (Part of) relations. In case of cross language, we will have SIM (similarity) relationship which relates concepts with the same meaning from different languages together.

3 Experiments

The experiments were conducted on INFILE test collection. The collection consists of 100,000 documents out of which 1597 relevant pair of documents were provided for evaluation purposes. Because we did not have access to a separate text collection, we build our KB using the same INFILE test collection. This may or may not introduce a bias into the experiments but on the other

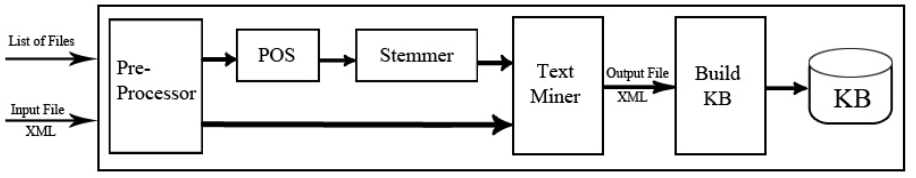


Fig. 2. Information Extractor Unit

hand what would be the benefit of testing our system with a KB with incompatible vocabulary. The KB only contains the relationships among the concepts and it does not contain any statistical information about the distributions of the concepts and their frequencies among the documents or topics. Therefore, the filtering hypothesis is not violated and the system is not able to make any assumption about the concepts and documents. In the rest of this section we describe different settings of the topic filtering process.

3.1 Concept Selection in Documents

Each document is treated as a query and is represented by a set of concepts $(Q_1, Q_2, ..., Q_k)$. These concepts are extracted from the heading and the content of the document. Only the concepts with a certainty more than the average threshold are used in document representation. Each concept Q_k is processed if $Freq_{Q_k} \geq 2$ and $\gamma_{Q_k} \geq avg$ where $avg = \sum \gamma_{Q_k}/N$. During the reasoning process new concepts will be generated and only those concepts that their certainty is bigger than the average certainty in the original document will be added to the representation and will be used in the later stages of reasoning for generating more concepts.

3.2 Evidence Combination

During the processes of reasoning multiple paths could be found between the concepts in a document and the concepts in a topic. Each path will have a certainty value which shows the system's confidence in that line of reasoning. These certainty values are combined in four steps to calculate the overall confidence in relevance of a document to a topic.

First level Combination: in each step of the reasoning if a concept is generated several times we keep only the one which has the maximum certainty value. In other words for $(inf_{i-t}, Q_{k-t}, topic_j, \gamma_j)$ take $max(\gamma)$.

Second Level Combination: many reasoning paths with the same type of inference or transform could relate a document concept to a topic concept. we only keep the path with maximum certainty value. In other words $(inf_{i-t}, Q_k, topic_j, \gamma_j)$ take $max(\gamma)$.

Third level Combination: for all the unique concepts in the document representation that have been related to the concepts in the topic through different

inferences $(inf_i, Q_k, topic_j, \gamma_j)$ calculate the sum of all the certainty values for all the paths connecting any document term to any topic term $(\sum \gamma - \prod \gamma)$ and return $(Q_j, topic_j, \gamma_j)$

Forth level Combination: for all unique concepts in the document representation that have been related to the concepts in topic query $(Q_j, topic_j, \gamma_j)$, calculate the sum of all the certainty values for all the paths connecting any document term to any topic term $(\sum \gamma - \prod \gamma)$ and return the list of retrieved documents $(topic_j, \gamma_j)$.

3.3 Topic Selection

The last component of our system is the Topic Filtering process. In this process, the system decides which one of the retrieved topics should be returned. We have conducted hundreds of experiments and have experimented with different factors that could influence this decision. One factor is the depth of the reasoning Process. During this process, the system traverses the concept hierarchies in the KB up and down to find new concepts that could be added to the document representation. Level indicates the number of levels that the system goes up and down the hierarchy using inference patterns. Level 0 means no inferences are applied on the concepts, i.e concepts have been matched against the topics directly. Another factor is M the number of topics we want to return for each document. $M = All$ means return all the topics that have been retrieved. We have experimented with $M = 1, 2, 3$ and All. Another factor is the confidence threshold. Some of the thresholds we have experimented with are:

No Threshold: no threshold means all certainty values are acceptable.

Threshold 1: $\gamma_{doc_j} \geq max(\gamma_{doc}) - \alpha * max(\gamma_{doc})$
In this case a topic is returned if its certainty value is within α percent of the maximum so far for that topic. The maximum is updated after each document.

Threshold 2: $\gamma_{doc_j} \geq avg(\gamma_{doc}) - \alpha * avg(\gamma_{doc})$
In this case, a topic is returned only if its certainty value is within α percent of the current average confidence value for that topic. Two different kinds of averages have been tried: A regular average so far and a monotonic average. The monotonic average is updating the average only by the increasing values of average.

Threshold 3: $\gamma_{doc_j} \geq min(\gamma_{doc}) - \alpha * min(\gamma_{doc})$
In this case a topic is returned only if its confidence value is within α percent of the minimum so far for that topic.

With 4 different M values and 3 levels and 5 different thresholds, we have conducted 141 different configurations. Each configuration has been tried with different values of α

4 Results

Table 2 shows the results for $Top1$ values of M with different levels and thresholds, with $\alpha = 0.7$ which gave the best results. In general, the min threshold

were the best threshold for certainty values. New documents were found at level 1 and 2 but not that many; so reasoning had a contribution when the level is either 1 or 2. It seems that using a certainty threshold is better than not using any thresholds. At level 1 for example, when $\alpha = 70$ is used, number of retrieved documents is dropped by 70 percent compared to when no threshold is used. This has increased the precision from 0.035 to 0.086. It seems that the similarity values of the relevant documents were below average but much higher than the minimum. This is an indication of a ranking problem.

Table 2. The results for $M = 1$ and $\alpha = 0.7$

Level	Threshold	Ret	Rel-Ret	Prec	Rec	$F-0.5$
2	$None$	19741	467	0.0347	0.2144	0.0532
	$\gamma_d \geq [max(\gamma) - 0.7 * max(\gamma)]$	11669	108	0.0405	0.1376	0.0553
	$\gamma_d \geq [\frac{\sum \gamma}{N}]$	6378	211	0.0562	0.0965	0.0571
	$\gamma_d \geq [min(\gamma) - 0.7 * min(\gamma)]$	6036	251	0.0742	0.1081	0.068
1	$None$	19616	461	0.0351	0.2114	0.0534
	$\gamma_d \geq [max(\gamma) - 0.7 * max(\gamma)]$	12326	313	0.0387	0.1377	0.0538
	$\gamma_d \geq [\frac{\sum \gamma}{N}]$	6469	230	0.0639	0.1029	0.063
	$\gamma_d \geq [min(\gamma) - 0.7 * min(\gamma)]$	5931	239	0.086	0.1073	**0.073**
0	$None$	17074	463	0.0409	0.2148	0.06
	$\gamma_d \geq [max(\gamma) - 0.7 * max(\gamma)$	14528	421	0.043	0.199	0.061
	$\gamma_d \geq [\frac{\sum \gamma}{N}]$	6170	242	0.0608	0.1316	0.0661
	$\gamma_d \geq [min(\gamma) - 0.7 * min(\gamma)]$	3937	198	0.0922	0.0923	0.0676

Table 3 shows the relationship among a number of inferences that has been used and a number of relevant documents retrieved. The number of inferences is an indirect indication of the length of the inference path. One inference means depth of $level0$ and direct match. However, since we have only levels $0, 1$ and 2 that means in each of these levels topics were retrieved through multiple inference paths. Basically, the more inference we go through the higher precision is we get but most of relevant retrieved topics were found in direct matching.
Table 4 shows how the number of matching terms between document and topic, are related to their precision. From Table 4, we observe that the more terms matches the higher the precision.

Based on the above observations we have run more experiments by combining the number of inference paths, the number of terms and certainty into a retrieval criteria. A few of these runs have been depicted in Table 5. Combining the number of terms matched with the number of inference paths or number of terms and certainty threshold has resulted in better performance and decreasing the number of retrieved documents. Although *precision* and *recall* and $F-measure$ are very important evaluation factors but showing as few documents to user

Table 3. Number of Inference paths and precision

Num of Inference paths	Ret	Rel-Ret	Prec
1	36801	290	0.007
2	7833	123	0.015
3	1944	100	0.051
4	584	34	0.058
5	138	16	0.115
6	55	11	0.2
7	20	4	0.2
8	11	4	0.36
9	5	1	0.2
10	1	0	0.0

Table 4. Number of terms matched between documents and topics and precision

Num of Terms	Ret	Rel-Ret	Prec
1	44400	343	0.007
2	2468	153	0.061
3	425	60	0.141
4	65	18	0.276
5	15	9	0.6
6	1	1	1

as possible also is important and improves user satisfaction in real systems. However, from these experiments it seems that there is an information overlap between the number of terms and the number of inferences which limits the usefulness of the approach.

Table 5. Number of Inference paths and precision

Num of matching terms	Num of inference paths	Certainty threshold	Rel-Ret	Ret	Prec	Rec	$F-0.5$
> 1	> 2		244	4054	0.061	0.153	**0.088**
> 1		> 0.3	304	5485	0.056	0.191	**0.087**
< 1 or	> 1		294	10591	0.028	0.185	0.049

A major problem we noticed in expansion of concepts is the lack of sufficient relations in KB. Also we felt the concepts weights generated from text processing, are not good representative of their value for the documents. Our conclusion from all these experiments was that, we need to investigate more text processing aspects and create better knowledge base. Once a better KB is built, we can work on certainty calculations and evidence combination. Also more sophisticated thresholds need to be experimented for both term and topic filtering.

5 Conclusion

We have built a system which uses inferences of the theory of Human Plausible Reasoning to infer new terms for expanding document representation and the relevance of a document to a topic in a filtering environment. We represent all the concepts regardless of their language of origin in the same knowledge base. Therefore the same inferences can retrieve any topic from any language

in response to arrival of a document. We used English INFILE text collection to build our KB and then we conducted hundreds of different experiments with different configurations. The recall of the system is less than what we expected and we can trace this back to the text processing unit and specially to the Text Miner unit. In future, we need to work on three aspects of our system, text processing, certainty calculations and evidence combination. Specially, we need to improve the quality of relations, we extract as they have a direct effect on recall of the system.

Acknowledgments. We would like to thank University of Wollongong in Dubai specially Information Technology and Telecommunication Services (ITTS) department as well as Dr Sherief Abdellah for providing us with a server to run our experiments.

References

1. Collins, A., Michalsky, R.: The Logic Of Plausible Reasoning A Core Theory. Cognitive Science 13, 1–49 (1989)
2. Collins, A., Burstein, M.: Modeling A Theory Of Human Plausible Reasoning. Artificial Intelligence III (2002)
3. Darrudi, E., Rahgozar, M., Oroumchian, F.: Human Plausible Reasoning for Question Answering Systems. In: International Conference on Advances in Intelligent Systems - Theory and Applications AISTA 2004 (2004)
4. Oroumchian, F., Arabi, B., Ashouri, E.: Using Plausible Inferences and Dempster-Shafer Theory of Evidence for Adaptive Information Filtering. In: 4th International Conference on Recent Advances in Soft Computing, pp. 248–254 (2002)
5. Karimzadehgan, K., Habibi, J., Oroumchian, F.: Logic-Based XML Information Retrival for Determining the Best Element to Retrive. In: Fuhr, N., Lalmas, M., Malik, S., Szlávik, Z. (eds.) INEX 2004. LNCS, vol. 3493, pp. 88–99. Springer, Heidelberg (2005)
6. Karimzadehgan, M., Belford, G., Oroumchian, F.: Expert Finding by Means of Plausible Inferences. In: International Conference on Information and Knowledge Engineering, IKE 2008, pp. 23–29 (2008)
7. Hugo, L.: MontyLingua: An end-to-end natural language processor with common sense (2004), http://web.media.mit.edu/~hugo/montylingua
8. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)

Hossur'Tech's Participation in CLEF 2009 INFILE Interactive Filtering

John Anton Chrisostom Ronald, Aurélie Rossi, and Christian Fluhr

Cadege/geol Semantics, 32. Rue Brancion,75015 Paris, France
{Ronald.chrisostom,rossi.aurelie,christian.fluhr}@gmail.com

Abstract. This paper describes the participation of our company formerly named Cadege / Hossur Tech and called now Geol Semantics in the task of filtering interactive CLEF 2009 INFILE and enhancements added after the experiment.

The Interactive filtering is something different from traditional information retrieval systems. In CLEF 2009 INFILE adaptive filtering task we have only the knowledge about the 50 different topics which are used as queries and nothing about the input corpus to filter. Documents are received and filtered one by one.

The fact that we know nothing about the corpus of the documents to filter, we were forced to use a linguistic approach for this filtering task. We have performed two CLEF 2009 INFILE interactive filtering French to French and French to English tasks, based on a deep linguistic process by using our own linguistic dictionaries.

ACM categories and subject descriptors: H.3.3 Information Search and Retrieval, Information filtering

1 Introduction

As Geol Semantics started from scratch in mid January 2009 to build an information extraction system based on deep linguistic analysis, our main objective was to experiment our comparison methods on actual data to design our future system. We have decided to base our linguistic processing upon weighted finite state automata. For this purpose we are developing a language to build multilingual linguistic processing based on the openFST framework. The planning of this technology development was not compatible with the participation in CLEF 2009 INFILE. We were aware of CLEF 2009 INFILE participation and that the CLEF 2009 INFILE test data afterwards were very valuable means for designing and tuning our future system. We have decided to participate in CLEF 2009 INFILE with much less elaborated linguistic processing than our final information extraction system should have. The main objective was to focus on studying the linguistic comparison strategies, weighting of intersections and finding a threshold to discriminate relevant and irrelevant documents in a context where statistical discriminating tools cannot be applied.

2 Functional Diagram

Each topic had been processed by using a limited version of XFST (XEROX Finite-state software) with our own resources. Part of speech tagging and lemmatization

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 374–380, 2010.

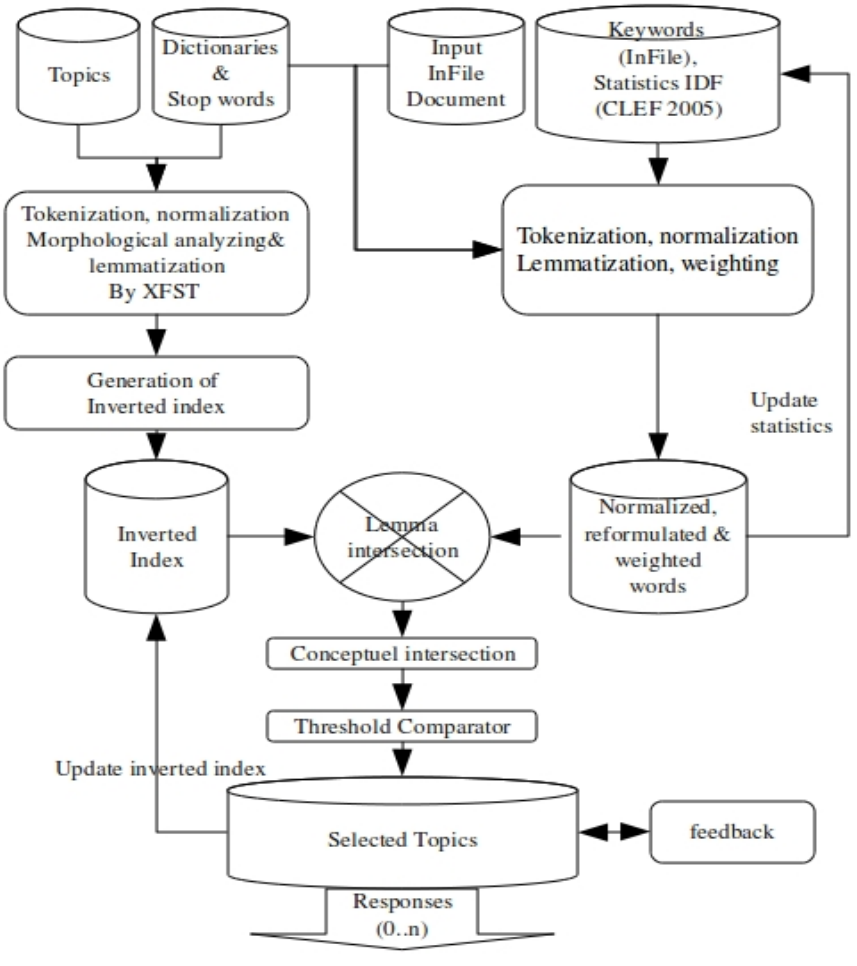


Fig. 1. Functional diagram

were obtained. For the input documents, it was unable to use the same linguistic processing because of volume limitation of our version of XFST, so only a simple dictionary look-up without disambiguation was used.

For each topic their title, description and narrative contents were used. The content of sample document was only used as a first positive feedback but not included strictly in the topic lemmatization. Only the content of the title and text were used.
We have generated all monolingual inferred equivalents from the whole words of input document for the French to French comparison and translation equivalents for the French to English Comparison.

A word intersection was computed and then a concept intersection was established. All words inferred from a same word were considered as representing a same concept.

Each concept contained in the topic-document intersection received a weight according to both a statistics computed on a similar corpus (Clef ad hoc corpus) and the fact that the concepts were in the topic keywords list or title or not. Proper nouns received also an increased weight.

A tentative threshold between relevant and irrelevant document was computed between the weight of the sample document and the maximum weight of the document relevant to other topics.

3 Linguistic Processing

A same linguistic processing must be used both on topics and input documents. In our case it was not possible. Waiting for our new technology, we had decided to use the Xerox XFST automaton compiler to develop a morpho-syntactic parsing based on our existing language resources. The available version of XFST had limitations that were not annoying to process limited amount of text like topic texts but prevented to process a large corpus of documents like the one used in CLEF 2009 INFILE.

For the topics, we had processed all the contents of the fields: title, descriptive, narrative and keywords using the full parser (part of speech tagging and compound recognition).

For the documents, we had processed headline and data content. As XFST could not be used we had developed a simple dictionary look-up giving all the possible lemmatization without disambiguation. As the dictionary used by XFST and the look-up for documents were the same, intersection could be obtained.

Before processing, accents and all punctuations including hyphens were removed. For the topics which were treated with accents, they were removed during the comparison processing.

4 The Problem of Interactive Adaptive Filtering

To categorize a document between two values (relevant and irrelevant) there are lot of methods generally based on a learning of positive and negative examples.

These methods could not be used in our case because at the beginning we had only one sample of a positive document per topic. And 200 feedbacks for 50 topics were not enough (sheared 4 feedbacks per topic) to have a good learning sample. In fact 4 feedbacks per topic is not a real life case. A user can provide more feedbacks and also we can infer simulated feedbacks by observing his actions on proposed documents.

We had to face up to two kinds of problems:

The first one was to elaborate a concept intersection between topics words and arriving input documents words and to weight the intersected concepts to get a relevant weight.

The second one was to choose a threshold under what, documents were considered as irrelevant.

The selection of this threshold provided us some difficulties; the fact of to choose a high threshold can lose relevant documents. Choosing a low threshold can provide us lot of irrelevant documents which is difficult to be corrected by only 4 feedbacks per topic.

5 Choices for Our Runs

5.1 Computation of the Word Intersection

All document words inferred equivalents in the same language using synonyms and other monolingual thesauri like reformulation rules for French to French Comparison.

Example: Lutte → combat, bataille, dispute.

For the word "Lutte" which means fight in French we have obtained "combat" same in English, "Bataille" battle in French and "dispute" same in English.

All document words inferred equivalent in the other language using bilingual reformation rules for French to English comparison.

Example: fight → combat, battle and dispute.

All lemmatized topic words were organized into an inverted file. A special procedure had computed the best intersection between the arriving input document and all the topics.

5.2 Weighting of the Words

All words do not provide us the same information (discriminative power). As it was not possible to obtain a statistic on a corpus which is not yet received, we have computed a general weight (based on inverted document frequency measure) on a similar corpus from ad hoc track of CLEF 2005.

We have also considered the importance of keywords and gave them a better weight than other topic words.

We have given also an increased weight for proper names. This was a conclusion of CEA (Commissariat à l'Energie Atomique, is French gouvernment funded technological research organisation.) in TREC 7, increasing the weight of proper names; enhance the performance of the filtering system. In fact, filtering systems are often used to track persons, companies or places.

5.3 Weighting of the Intersection

Topic words and their inferred words did not put together a big set of words. Links between original topic words and inferred ones were kept. We will now consider not the word intersection but the concept intersection. The original topic words and all the words inferred from it were considered as equivalent to represent a concept. The weight attributed to a concept was the minimum of the all words representing the concept. Weights were added to provide a relevant value to the intersection (i.e. to the topic).

6 Computation of a First Threshold

As we have a relevant document in the topic, we computed the value of its intersection with the topic. This provided an upper value for the threshold.

To obtain a sample of irrelevant documents we have considered that all the sample documents attached to other topics are irrelevant documents. In some cases when topics have some intersection between each other, this can be a good way to discriminate topics.

We have chosen the greater value for the irrelevant example as the lowest limit for the threshold.

The Threshold used at the beginning was set at lower value +85% of the difference between lower and upper threshold.

7 Adaptations

Three kinds of adaptation had been used:

The first concerns the threshold. If a positive feedback is provided and at the same time the conceptual intersection value is lower than the previous upper threshold, the new one is used. If a negative feedback is given and the value is upper than the previous lower threshold, the new one is used.

The second adaptation is devoted to add relevant vocabulary to the topic. As the fact that a word is in one relevant document is not a strong reason to consider it as a relevant word for the topic, we have considered, to eliminate hazard, that the presence of a word in two documents attested as relevant is necessary to add it into the topic word set.

The weights of words (based on IDF computation) are updated... new words are added by the processing of entering document.

8 Results and Discussion

We cannot really compare our results with others teams results, because we were only two teams to participate in CLEF 2009 INFILE adaptive filtering task. The other team has used monolingual English to English techniques while we have used the cross lingual French to English approach and they have not processed the full data.

Table 1. Results of interactive filtering

Run	Precision	Recall	F-Measure	Utility	Anticipation
Cross lingual French-English	0.10	0.45	0.10	0.07	0.59
Monolingual French-French	0.05	0.31	0.06	0.05	0.53

There were more participants in the batch task than the adaptive filtering task but it was also monolingual English to English runs. Our results in French to English in adaptive task were comparable with their results.

We have noticed that our bilingual English-French interactive filtering task's results were pretty good compared to monolingual French-French interactive filtering task's results.

So we have paid more attention to improve the French to French results.

According to the precision value, our filtering system's bandwidth was very large so we have not filtered very well noisy input documents , to bring our filter's bandwidth to a narrow acceptable value, we have established a population study of concepts in the input document.

The population study is based on a computation of the concentration of concepts in a limited zone, here a limited number of words from the input documents.

This study was slightly beneficial because we have improved the value of our precision. Finally we have removed the monolingual reformulation from our system then we have better results.

Table 2. Results after CLEF 2009 INFILE

Run	Precision	Recall	F-Measure	Utility	Anticipation
CLEF 2009 INFILE	0.05	0.31	0.06	0.05	0.53
With Population studies	0.12	0.22	0.11	0.16	0.35
Population studies without reformulation	0.23	0.26	0.15	0.23	0.37

These results helped us to detect our problem of precision loose, for monolingual French – French adaptive filtering task. We have increased the precision of monolingual French to French. We think the population study must provide better results on cross lingual French-English Task also.

9 Conclusions

To sum up, in this paper, we have described our participation of CLEF 2009 INFILE. We have added an extra component in order to reduce the noise from the input documents by analyzing words population.

We have also reduced the noise provided from our French reformulation dictionary by simply not using the component of reformulation. This do not mean that monolingual reformulation must not be used but this means that in our monolingual reformulation dictionary there are more relations decreasing the precision than relations bringing a better recall. A detailed study of this dictionary content and its influence on precision and recall is necessary. These two strategies are quite successful; we have enhanced the performance of our French to French monolingual filtering system.

Although, our results are better than the end of CLEF 2009 INFILE, we are still working on it to enhance our filtering performances. Because of this difference, the results obtained in cross language retrieval are more in line with the new monolingual one.

Now to improve our results, we need to include the domain knowledge which is necessary to process queries where this knowledge is compulsory like for extending the concept of "international sport competition" by "Olympic games, European cup, world cup" and so on "doping drug" by the names of the drug. We intend to build up this domain knowledge by extracting it for the web. For example if you ask to a well known search engine the following query "doping drug list" you obtain as second document:" THE 2009 PROHIBITED LIST INTERNATIONAL STANDARD » which provides the full list of prohibited drugs.

References

1. Bisson, F., Charron, J., Fluhr, C., Schmit, D.: EMIR at the CLIR track of TREC7, November 9-11, Gaithersburg, MD,USA,(1998)
2. The TREC 2001 Filtering Track Report, Stephen Robertson (Microsoft Research Cambridge,Uk), Ian Soboroff (NIST, USA), Gaitherburg, MD,USA (November 2001)
3. Besançon, R., Chaudiron, S., Mostefa, D., Timimi, I., Choukri, K.: The InFile project: a crosslingual filtering systems evaluation campaign. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, Springer, Heidelberg (2009)

Experiments with Google News for Filtering Newswire Articles

Arturo Montejo-Ráez, José M. Perea-Ortega,
Manuel Carlos Díaz-Galiano, and L. Alfonso Ureña-López

SINAI Research Group*, Computer Science Department, University of Jaén,
Campus Las Lagunillas, Edificio A3, E-23071, Jaén, Spain
{amontejo,jmperea,mcdiaz,laurena}@ujaen.es

Abstract. This paper describes an approach based on the use of Google News[1] as a source of information in order to generate a learning corpus for an information filtering task. The INFILE (INformation FILtering Evaluation) track of the CLEF (Cross-Lingual Evaluation Forum) 2009 campaign has been used as framework. The information filtering task can be seen as a document classification task, so a supervised learning scheme has been followed. Two learning corpora have been proved: one using the text of the topics as learning data to train a classifier, and another one where training data have been generated from Google News pages, using the keywords of topics as queries. Results show that the use of Google News for generating learning data does not improve the results obtained using only topic descriptions as learning corpora.

1 Introduction

INFILE is a new track within the CLEF campaign that was proposed as pilot track in 2008 [1]. Its purpose is to evaluate crosslingual adaptive filtering systems [2]. Information filtering in the INFILE track is considered in the context of competitive intelligence: in this context, the evaluation protocol of the campaign has been designed with a particular attention to the context of use of filtering systems by real professional users [3].

INFILE proposes two main tasks: the adaptive filtering task already proposed in 2008 [1] and the new one proposed in 2009 about testing batch filtering systems [3]. Both tasks are crosslingual: English, French and Arabic are available for the documents and topics. The collection contains documents to be classified according to 50 different topics, but no training samples are supplied. Each of the 50 topics is described in a file with short a description text, a set of related keywords and some other fields.

The experiments presented in this paper are based only on the batch filtering task and using solely English texts. As a supervised learning approach is followed, two learning corpora have been generated:

* http://sinai.ujaen.es
[1] http://news.google.com/

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 381–384, 2010.

- one using the text of the topics as learning data to train a classifier (referred as *topics descriptions*),
- and another one where training data has been generated from Google News pages, using the keywords from topics as queries (referred as *Google News* corpus).

The web has been extensively used as resource when dealing with different text mining problems [4,6]. We have applied it in video retrieval and classification [5,8].

This paper is organized as follows: Section 2 describes the approach followed in this work. Then, in Section 3, experiments and results are shown. Finally, in Section 4, the conclusions and further work are presented.

2 System Description

A traditional supervised learning scheme has been followed to solve the filtering task. The difference between the two experiments submitted relies on the training corpus used. One was on Google News entries and another on the descriptions of the topics provided. These corpora have served as learning data for building a model which was, thus, used for classifying every document into one of the fifty different classes proposed. The learning algorithm was Support Vector Machines (SVM) [7] on both experiments.

The Google News corpus was generated by querying Google News on each of the topics keywords. If a topic is used as query in the Google News search engine, the resulting list of web documents can be considered as representative samples for that topic. This is the idea behind this corpus, where a total of 50 documents per keyword in a topic were downloaded. The procedure for obtaining the web documents was simple. The following URL was used to ask Google News for each keyword:

```
http://news.google.com/news/search?hl=en&num=50&q=$keyword
```

Then, the returned http links were followed and retrieved as documents for this corpus. For example, topic 101 contains the keywords *doping*, *legislation doping*, *athletes*, *doping substances* and *fight against doping*. For each of these keywords, 50 links were retrieved, downloaded and their HTML cleaned out. In this way, about 250 documents existed per topic.

For each learning corpus generated, a SVM model was trained on it. This is a binary classifier turned into a multi-class classifier by training a different SVM model per topic. The topic with the highest confidence was selected as label for the incoming document and, therefore, the document was routed to that topic. It is important to note here that a label was proposed for every one of all incoming documents, that is, no document was left without one of the 50 labels (topics). The results obtained have been compared to those where only provided texts on topics were used as learning data.

3 Experiments and Results

The experiments carried out in this paper are based on the batch filtering task of INFILE and using English as source and target languages. Two learning corpora have been used in the supervised scheme followed: topic descriptions and corpus generated from Google News. Evaluation scores for these experiments are presented in Table 1. `Num_rel` refers to the total number of relevant documents that were in the collection, that is, the number of documents that were actually classified in any of the available topics. `Num_rel_ret` refers to the total number of relevant (well classified) documents found by our system. It is important to note that, from the total of 100,000 documents in the collection, only 1,597 were related to a topic.

Table 1. Overall results for the experiments

Learning corpus	Num_rel	Num_rel_ret	Precision	Recall	F-score
Topics descriptions	1597	940	0.02	0.50	0.04
Google News	1597	196	0.01	0.08	0.01

The results obtained are discouraging: few relevant assignations are made. In fact, the use of Google News as learning source leads to very poor results. Both precision and recall are low, as can be seen in Table 1. Regarding the precision, the use of topic descriptions as learning corpus doubles the precision obtained using Google News as learning corpus. For recall, the improvement obtained using topic descriptions as learning corpus is overly broad: 0.42 points better than using Google News as learning corpus. Recall degradation is also clear when using Google News. In the case of topic descriptions, a much better value is obtained compared to the precision metric, that is, more documents were correctly routed.

4 Conclusions and Further Work

In this work, a supervised learning approach has been followed for solving the document filtering task. SVM has been chosen as learning algorithm. Two experiments have been carried out. The difference between both is the learning corpus used: topic descriptions and Google News.

Google News as a source of information for generating a learning corpus has shown quite bad results. After inspecting this problematic learning corpus, we found that huge amounts of useless text was not filtered. Therefore, we plan to improve the quality of the data extracted from the web in order to avoid undesirable side effects due to noisy content, i.e. web content (headers, footers, links to other sections...) not related to the new itself. Therefore, a deeper work on information extraction has to be performed.

Although the results obtained in this task are really very low in terms of performance, it represents a challenge in text mining, as real data has been used,

compared to previous too controlled corpora. We expect to continue our research on this data, and analyze in depth the effect of incorporating web content in filtering tasks.

Acknowledgments

This paper has been partially supported by a grant from the Spanish Government, project TEXT-COOL 2.0 (TIN2009-13391-C04-02), a grant from the Andalusian Government, project GeOasis (P08-TIC-41999), and a grant from the University of Jaén, project RFC/PP2008/UJA-08-16-14.

References

1. Besançon, R., Chaudiron, S., Mostefa, D., Hamon, O., Timimi, I., Choukri, K.: Overview of CLEF 2008 INFILE Pilot Track. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 939–946. Springer, Heidelberg (2009)
2. Besançon, R., Chaudiron, S., Mostefa, D., Timimi, I., Choukri, K.: The INFILE Project: a Crosslingual Filtering Systems Evaluation Campaign. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008). European Language Resources Association (ELRA) (2008)
3. Besançon, R., Chaudiron, S., Mostefa, D., Timimi, I., Choukri, K., Laïb, M.: Overview of CLEF 2009 INFILE track. In: Peters, C., Nunzio, G.D., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.) In Press. LNCS, Springer, Heidelberg (2009)
4. Couto, F.M., Martins, B., Silva, M.J.: Classifying biological articles using web resources. In: SAC 2004, Proceedings of the 2004 ACM symposium on Applied computing. pp. 111–115. ACM, New York (2004)
5. Díaz-Galiano, M.C., Perea-Ortega, J.M., Martín-Valdivia, M.T., Montejo-Ráez, A., Ureña-López, L.A.: SINAI at TRECVID 2007. In: Over, P. (ed.) Proceedings of the TRECVID 2007 Workshop (TRECVID 2007) (2007)
6. Gligorov, R., ten Kate, W., Aleksovski, Z., van Harmelen, F.: Using google distance to weight approximate ontology matches. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, pp. 767–776. ACM, New York (2007)
7. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998), citeseer.ist.psu.edu/joachims97text.html
8. Perea-Ortega, J.M., Montejo-Ráez, A., Díaz-Galiano, M.C., Martín-Valdivia, M.T., Ureña-López, L.A.: Using an Information Retrieval System for Video Classification. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 927–930. Springer, Heidelberg (2009)

CLEF-IP 2009: Retrieval Experiments in the Intellectual Property Domain

Giovanna Roda[1], John Tait[2], Florina Piroi[2], and Veronika Zenz[1]

[1] Matrixware Information Services GmbH
[2] The Information Retrieval Facility (IRF)
Vienna, Austria
{g.roda,v.zenz}@matrixware.com,
{j.tait,f.piroi}@ir-facility.org

Abstract. The CLEF–IP track ran for the first time within CLEF 2009. The purpose of the track was twofold: to encourage and facilitate research in the area of patent retrieval by providing a large clean data set for experimentation; to create a large test collection of patents in the three main European languages for the evaluation of cross–lingual information access. The track focused on the task of prior art search. The 15 European teams who participated in the track deployed a rich range of Information Retrieval techniques adapting them to this new specific domain and task. A large-scale test collection for evaluation purposes was created by exploiting patent citations.

1 Introduction

The Cross Language Evaluation Forum CLEF[1] originally arose from a work on Cross Lingual Information Retrieval in the US Federal National Institute of Standards and Technology Text Retrieval Conference TREC[2] but has been run separately since 2000. Each year since then a number of tasks on both cross–lingual information retrieval (CLIR) and monolingual information retrieval in non–English languages have been run. In 2008 the Information Retrieval Facility (IRF) and Matrixware Information Services GmbH obtained the agreement to run a track which allowed groups to assess their systems on a large collection of patent documents containing a mixture of English, French and German documents derived from European Patent Office data. This became known as the CLEF–IP track, which investigates IR techniques in the Intellectual Property domain of patents.

One main requirement for a patent to be granted is that the invention it describes should be novel: that is there should be no earlier patent or other publication describing the invention. The novelty breaking document can be published anywhere in any language. Hence when a person undertakes a search, for example to determine whether an idea is potentially patentable, or to try to

[1] http://www.clef-campaign.org
[2] http://trec.nist.gov

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 385–409, 2010.

prove a patent should not have been granted (a so-called opposition search), the search is inherently cross–lingual, especially if it is exhaustive.

The patent system allows inventors a monopoly on the use of their invention for a fixed period of time in return for public disclosure of the invention. Furthermore, the patent system is a major underpinning of the company value in a number of industries, which makes patent retrieval an important economic activity.

Although there is important previous academic research work on patent retrieval (see for example the ACM SIGIR 2000 Workshop [11] or more recently the NTCIR workshop series [6], there was little work involving non–English European Languages and participation by European groups was low. CLEF–IP grew out of desire to promote such European research work and also to encourage academic use of a large clean collection of patents being made available to researchers by Matrixware (through the Information Retrieval Facility).

CLEF–IP has been a major success. For the first time a large number of European groups (15) have been working on a patent corpus of significant size within an integrated and single IR evaluation collection. Although it would be unreasonable to pretend the work is beyond criticism it does represent a significant step forward for both IR community and patent searchers.

The usual outcome of an evaluation track is a test collection that can be used for comparing retrieval techniques. A test collection traditionally consists of three sets of data:

- target data
- topics
- relevance assessments

Relevance assessments are subsets of the target data set that fulfill the information needs represented by the topics. They are relative to a well–defined task or *information need*. The main task in CLEF–IP was to find *prior art* for a given patent. This task is in principle similar to an ad–hoc search but its name was motivated by the fact that relevance assessments used by the track consisted in the patent documents cited as prior art in search reports. This method for automatically generating test collections for patent retrieval is described in [7]. Notwithstanding the bias introduced by this particular choice of relevance assessments that will require some additional investigation, having a large number of such assessments available made it possible to carry out an automatic evaluation on a reasonably large set of topics. Manual assessments are costly and especially for patents they are not only time–consuming but they require the specialized domain expertise that only a professional patent searcher has.

While deciding on how to generate relevance assessments, we also considered the possibility of exploiting some social assessment platform such as the one launched by the USPTO PeerToPatent project[3] and endorsed by the Linux foundation's OSAPA initiative[4]. Such an experimental approach for obtaining

[3] http://www.peertopatent.org

[4] http://www.linuxfoundation.org/programs/legal/osapa

relevance assessments did not fit into the track's time frame, however it might be reconsidered as a viable alternative in the future.

Having prior art citations available as relevance assessments for a large number of patents allowed us to concentrate on some criteria for the selection of the topics to be used in the final collection. These criteria were of statistical nature as well as concerning the quality of the patent documents available. The considerations that led to the final choice of topics are described in [15].

The CLEF–IP target data set where prior art documents were to be found consisted in approximately 2 million patent documents corresponding to 1 million individual patents. The set was restricted to European patents because they provide a good assortment of languages. In fact, any patent that is granted by the European Patent Office has to include a translation of the claims in all three official EPO languages (English, German, and French). The choice of granted patent documents as topics was also motivated by the fact that they would provide a parallel corpus in three languages on which to compare retrieval effectiveness for different languages. This choice has later been contested by patent experts who associate prior art searches with patent applications and not with granted patents. Therefore, future tracks will have patent application documents as targets for the prior art search task.

The number of participants is good for a newly established track and it shows that there is a rising interest in the IR community in tackling the challenge of patent search. Patent search is a task that requires a deep know–how about the patent domain and it is clear that the more the patent system is understood the better this know–how can be translated into automated retrieval strategies. Track participants were faced among others with questions such as:

- which textual fields to index (title, abstract, claims, description)?
- out of which fields to extract query terms?
- how to select or generate query terms?
- how to exploit metadata (classification codes, authors, citations)?

In addition to the new search task, the large amount of data—not customary for a CLEF track—constituted an additional challenge for the participating teams and stimulated them to consider performance issues for their implementations.

A wide range of Information Retrieval techniques was adopted by the participating teams for tackling the prior art search task. By looking at the results it is evident that an in–depth knowledge of the patent domain and of the patent search process can be translated into highly customized retrieval strategies that lead to optimal results. The most interesting mixture of techniques—meant to mimic a real patent searching process—was implemented by the Humboldt University's team and we believe that their approach will be a model for next year's competition. Some of the most successful retrieval strategies adopted by track's participants are the use of non–textual elements of patents provided in form of meta–data (e. g. classification codes, priority information), machine learning techniques and enrichment of the text data with the help of thesauri and concept databases. Multi–linguality was not the main focus of participants for the first year. This maybe partly due to the fact that this aspect of the search was

not stressed enough. A different balance of languages in next year's topics might encourage participants to focus on the use of multi–lingual indexing and translations of query terms, which we believe will lead to an improvement of overall results.

Being aware that a single query–and–result set will by no means be representative of a typical patent search process, we adopted for evaluation standard measures such as precision, recall, MAP, and nDCG. Additionally, we carried out some investigations on the significance of the experimental results obtained from our evaluation.

In addition to the automatic evaluation, we had a dozen topics manually assessed by patent experts. The fact that prior art citations are relative to patent applications while queries were generated from documents corresponding to granted patents generated some ambiguities and we could not exploit the results obtained by this second type of evaluation. However, the interaction with the patent experts community improved over time and brought to a mutual understanding of methods and goals that will be useful in defining future evaluation tasks.

Structure of the paper. Section 2 describes in detail target data (2.1), tasks and topics (2.2), and how relevance assessments were obtained (2.3). The generation of relevance assessments for the CLEF–IP test collection was done in a completely automatic fashion using the prior art items (citations) found in search reports. This approach allowed us to produce a test collection with a large number of topics that has many advantages for experimenting and comparing results. The limits of this methodology and some suggestions on how to generalize it in forthcoming tracks are discussed in Section 5.

Section 3 deals with participants, runs submitted and techniques adopted for the search task. After a short description of the Alfresco-based submission system (3.1), in 3.2 we list participants and the experiments they submitted summarizing their participants' contributions and providing a catalogue of all techniques adopted for the track.

Section 4 deals with the evaluation results. In 4.1 measures used and summary result tables are presented. We did some additional analysis on the evaluation results: correlation among different size bundles (4.2) and some consideration on statistical significance (4.2) of our measurings.

Finally, in Section 5 we discuss lessons learned and plans for future tracks and conclude the paper with an epilogue (Section 6).

2 The CLEF-IP Patent Test Collection

2.1 Document Collection

The CLEF–IP track had at its disposal a collection of patent documents published between 1978 and 2006 at the European Patent Office (EPO). The whole collection consists of approximately 1.6 million individual patents. As suggested in [7], we split the available data into two parts

1. the **test collection corpus** (or target dataset)—all documents with publication date between 1985 and 2000 (1,958,955 patent documents pertaining to 1,022,388 patents, 75GB);
2. the **pool for topic selection**—all documents with publication date from 2001 to 2006 (712,889 patent documents pertaining to 518,035 patents, 25GB).

Patents published prior to 1985 were excluded from the outset, as before this year many documents were not filed in electronic form and the optical character recognition software that was used to digitize the documents produced noisy data. The upper limit, 2006, was induced by our data provider—a commercial institution—which, at the time the track was agreed on, had not made more recent documents available.

Patent documents, provided in the XML format, are structured documents consisting of four major sections: bibliographic data, abstract, description and claims. Non–linguistic parts of patents like technical drawings, tables of formulas were left out which put the focus of this years track on the (multi)lingual aspect of patent retrieval: EPO patents are written in one of the three official languages English, German and French. 69% of the documents in the CLEF–IP collection have English as their main language, 23% German and 7% French. The claims of a granted patent are available in all 3 languages and also other sections, especially the title are given in several languages. That means the document collection itself is multilingual, with the different text sections being labeled with a language code.

Patent documents and kind codes. In general, to one patent are associated several patent documents published at different stages of the patent's life–cycle. Each document is marked with a *kind code* that specifies the stage it was published in. The kind code is denoted by a letter possibly followed by a one–digit numerical code that gives additional information on the nature of the document. In the case of the EPO, "A" stands for a patent's application stage and "B" for a patent's granted stage, "B1" denotes a patent specification and "B2" a later, amended version of the patent specification[5].

Characteristic to the CLEF–IP patent document collection is that files corresponding to patent documents published at various stages need not contain the whole data pertinent to a patent. For example, a "B1" document of a patent granted by the EPO contains, among other, the title, the description, and the claims in three languages (English, German, French), but it usually does not contain an abstract, while an "A2" document contains the original patent application (in one language) but no citation information except the one provided by the applicant.[6]

[5] For a complete list of kind codes used by various patent offices see http://tinyurl.com/EPO-kindcodes

[6] It is not in the scope of this paper to discuss the origins of the content in the EPO patent documents. We only note that applications to the EPO may originate from patents granted by other patent offices, in which case the EPOmay publish patent documents with incomplete content, referring to the original patent.

The CLEF–IP collection was delivered to the participants "as is", without joining the documents related to the same patent into one document. Since the objective of a search are patents (identified by patent numbers, without kind code), it is up to the participants to collate multiple retrieved documents for a single patent into one result.

2.2 Tasks and Topics

The goal of the CLEF–IP tasks consisted in finding prior art for a patent. The tasks mimic an important real–life scenario of an IP search professional. Performed at various stages of the patent life–cycle, prior art search is one of the most common search types and a critical activity in the patent domain. Before applying for a patent, inventors perform a such a search to determine whether the invention fulfills the requirement of novelty and to formulate the claims as to not conflict with existing prior art. During the application procedure, a prior art search is executed by patent examiners at the respective patent office, in order to determine the patentability of an application by uncovering relevant material published prior to the filing date of the application. Finally parties that try to oppose a granted patent use this kind of search to unveil prior art that invalidates patents claims of originality.

For detailed information on information sources in patents and patent searching see [3] and [9].

Tasks. Participants were provided with sets of patents from the topic pool and asked to return all patents in the collection which constituted prior art for the given topic patents. Participants could choose among different topic sets of sizes ranging from 500 to 10000.

The general goal in CLEF–IP was to find prior art for a given topic patent. We proposed one main task and three optional language subtasks. For the language subtasks a different topic representation was adopted that allowed to focus on the impact of the language used for query formulation.

The main task of the track did not restrict the language used for retrieving documents. Participants were allowed to exploit the multilinguality of the patent topics.

The three optional subtasks were dedicated to cross–lingual search. According to Rule 71(3) of the European Patent Convention [1], European granted patents must contain claims in the three official languages of the European Patent Office (English, French, and German). This data is well–suited for investigating the effect of languages in the retrieval of prior art. In the three parallel multi–lingual subtasks topics are represented by title and claims, in the respective language, extracted from the same "B1" patent document. Participants were presented the same patents as in the main task, but with textual parts (title, claims) only in one language. The usage of bibliographic data, e.g. IPC classes was allowed.

Topic representation. In CLEF–IP a topic is itself a patent. Since patents come in several version corresponding to the different stages of the patent's life–cycle, we were faced with the problem of how to best represent a patent topic.

A patent examiner initiates a prior art search with a full patent application, hence one could think about taking highest version of the patent application's file would be best for simulating a real search task. However such a choice would have led to a large number of topics with missing fields. For instance, for EuroPCTs patents (currently about 70% of EP applications are EuroPCTs) whose PCT predecessor was published in English, French or German, the application files contain only bibliographic data (no abstract and no description or claims).

In order to overcome these shortcomings of the data, we decided to assemble a virtual "patent application file" to be used as a topic by starting from the "B1" document. If the abstract was missing in the B1 document we added it from the most current document where the abstract was included. Finally we removed citation information from the bibliographical content of the patent document.

Topic selection. Since relevance assessments were generated by exploiting existing manually created information (see section 2.3) CLEF–IP had a topic pool of hundreds of thousands of patents at hand. Evaluation platforms usually strive to evaluate against large numbers of topics, as robustness and reliability of the evaluation results increase with the number of topics [18] [19]. This is especially true when relevance judgments are not complete and the number of relevant documents per topic is very small as is the case in CLEF–IP where each topic has on average only 6 relevant documents. In order to maximize the number of topics while still allowing also groups with less computational resources to participate, four different topic bundles were assembled that differed in the number of topics. For each task participants could chose between the topics set S (500 topics), M (1,000 topics), L (5,000 topics), and XL (10,000 topics) with the smaller sets being subsets of the larger ones. Participants were asked to submit results for the largest of the 4 sets they were able to process.

From the initial pool of 500,000 potential topics, candidate topics were selected according to the following criteria:

1. availability of granted patent
2. full text description available
3. at least three citations
4. at least one highly relevant citation

The first criteria restricts the pool of candidate topics to those patents for which a granted patent is available. This restriction was imposed in order to guarantee that each topic would include claims in the three official languages of the EPO: German, English and French. In this fashion, we are also able to provide topics that can be used for parallel multi–lingual tasks. Still, not all patent documents corresponding to granted patents contained a full text description. Hence we

imposed this additional requirement on a topic. Starting from a topics pool of approximately 500,000 patents, we were left with almost 16,000 patents fulfilling the above requirements. From these patents, we randomly selected 10,000 topics, which bundled in four subsets constitute the final topic sets. In the same manner 500 topics were chosen which together with relevance assessments were provided to the participants as training set.

For an in–depth discussion of topic selection for CLEF–IP see [15].

2.3 Relevance Assessment Methodology

This section describes the two types of relevance assessments used in CLEF–IP 2009: (1) assessments automatically extracted from patent citations as well as (2) manual assessments done by volunteering patent experts.

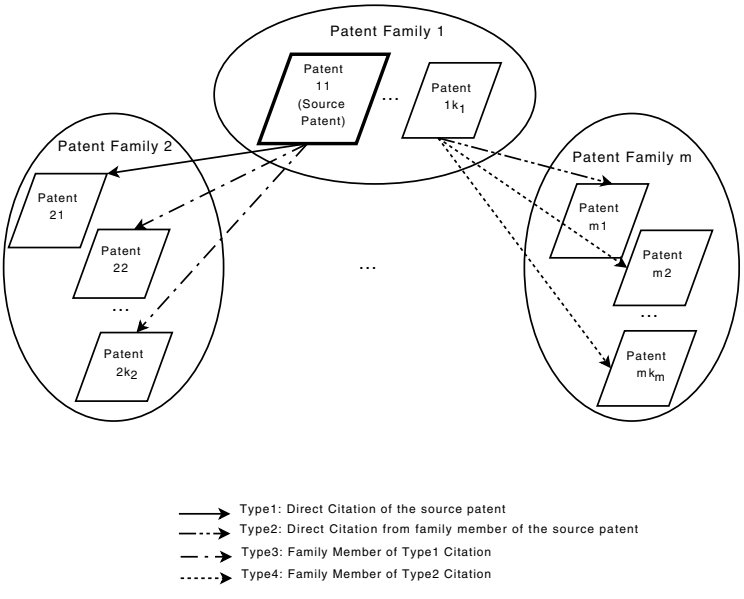


Fig. 1. Patent citation extension used in CLEF-IP09

Automatic Relevance Assessment. A common challenge in IR evaluation is the creation of ground truth data against which to evaluate retrieval systems. The common procedure of pooling and manual assessment is very labor–intensive. Voluntary assessors are difficult to find, especially when expert knowledge is required as is the case of the patent field. Researchers in the field of patents and prior art search, however, are in the lucky position of already having partial ground truth at hand: patent citations.

Citations are extracted from several sources:

1. applicant's disclosure : some patent offices (e.g. USPTO) require applicants to disclose all known relevant publications when applying for a patent;
2. patent office search report : each patent office will do a search for prior art to judge the novelty of a patent;
3. opposition procedures : often enough, a company will monitor granted patents of its competitors and, if possible, file an opposition procedure (i.e. a claim that a granted patent is not actually novel).

There are two major advantages of extracting ground truth from citations. First citations are established by members of the patent offices, applicants and patent attorneys, in short by highly qualified people. Second, search reports are publicly available and are made for any patent application, which leads to a huge set of assessment material that allows the track organizers to scale the set of topics easily and automatically.

Methodology. The general method for generating relevance assessments from patent citations is described in [7]. This idea had already been exploited at the NTCIR workshop series[7]. Further discussions at the 1st IRF Symposium in 2007 [8] led to a clearer formalization of the method.

For CLEF–IP 2009 we used an extended list of citations that includes not only patents cited directly by the patent topic, but also patents cited by patent family members and family members of cited patents. By means of patent families we were able to increase the number of citations by a factor of seven. Figure 1 illustrates the process of gathering direct and extended citations.

A *patent family* consists of patents granted by different patent authorities but related to the same invention (one also says that all patents in a family share the same *priority* data). For CLEF–IP this close (also called *simple*) patent family definition was applied, as opposed to the extended patent family definition which also includes patents related via a split of one patent application into two or more patents. Figure 1 (from [12]) illustrates an example of extended families.

In the process of gathering citations, patents from ~ 70 different patent offices (including USPTO, SIPO, JPO, etc.) were considered. Out of the resulting lists of citations all non–EPO patents were discarded as they were not present in the target data set and thus not relevant to our track.

Characteristics of patent citations as relevance judgments. What is to be noted when using citations lists as relevant judgments is that:

– citations have different degrees of relevancy (e.g. sometimes applicants cite not really relevant patents). This can be spotted easily by the citation labels as coming from applicant or from examiner. Patent experts advise to choose patents with less than 25–30 citations coming from the applicant.

[7] http://research.nii.ac.jp/ntcir/
[8] http://www.ir-facility.org/events/irf-symposium/2007

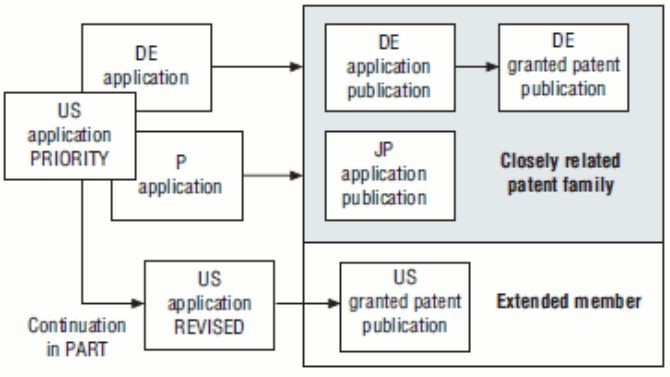


Fig. 2. Example for close and extended patent families. Source: OECD ([12]).

– the lists are incomplete: even though, by considering patent families and opposition procedures, we obtained fairly good lists of judgments, the nature of the search is such that it often stops when it finds one or only a few documents that are very relevant for the patent. The Guidelines for examination in the EPO [2] prescribe that if the search results in several documents of equal relevance, the search report should normally contain no more than one of them. This means that we have incomplete recall bases which must be taken into account when interpreting the evaluation results presented here.

Further automatic methods. We show here possible ways of extending the set of relevance judgements. These sources have not been used in the current evaluation procedure as they seem to be less reliable indicators of relevancy. Nevertheless they are interesting avenues to consider in the future, which is why they are mentioned here:

A list of citations can be expanded by looking at patents cited by the cited patents, assuming some level of transitivity of the 'citation' relation. It is however arguable how relevant a patent C is to patent A if we have something like *A cites B* and *B cites C*. Moreover, such a judgment cannot be done automatically.

In addition, a number of patent specific features can be used to identify potentially relevant documents: *co–authorship* (in this case "co-inventorship"), if we assume that an inventor generally has one area of research, *co–ownership* if we assume that a company specializes in one field, or *co–classification* if two patents are classified in the same class according to one of the different classification models at different patent offices.

Recently, a new approach for extracting prior art items from citations has been presented in [17].

Manual Relevance Assessment by Patent Experts. A number of patent experts were contacted to do manual assessments for a small part of the track's experimental results. Finding a commong language to relate the project's goals and procedures was not an easy task, nor was it motivating them to invest some time for this assessment activity. Nevertheless, a total of 7 experts agreed to assess the relevance of retrieved patents for one or more topics that were chosen out of our collection according to their area of expertise. A limit of around 200 retrieved patents to assess seemed to provide an acceptable amount of work. This limit allowed us to pool experimental data up to depth 20.

The engagement of patent experts resulted in 12 topics assessed up to rank 20 for all runs. A total of 3140 retrieval results were assessed with an average of 264 results per topic. In Section 4 we report on the results obtained by using this additional small set of data for evaluation.

3 Submissions

CLEF–IP relied on a Web–based system implemented on Alfresco for the submission of experiments. The 14 participating teams contributed a total of 70 experiments. In this section we summarize the work of the track's participants on the CLEF–IP challenge.

3.1 Submission System

Clear and detailed guidelines together with automated format checks are critical in managing large-scale experimentations.

For the upload and verification of runs a track management system was developed based on the open source document management system Alfresco[9] and the web interface Docasu[10]. The system provides an easy-to-use Web-frontend that allows participants to upload and download runs and any other type of file (e.g. descriptions of the runs). The system offers version control as well as a number of syntactical correctness tests. The validation process that is triggered on submission of a run returns a detailed description of the problematic content. This is added as an annotation to the run and is displayed in the user interface. Most format errors were therefore detected automatically and corrected by the participants themselves. Still one error type passed the validation and made the postprocessing of some runs necessary: patents listed as relevant on several different ranks for the same topic patent. Such duplicate entries were filtered out by us before evaluation.

3.2 Description of Submitted Runs

A total of 70 experiments from 14 different teams and 15 participating institutions (the University of Tampere and SICS joined forces) was submitted to

[9] http://www.alfresco.com/
[10] http://docasu.sourceforge.net/

Table 1. List of active participants and runs submitted

Id	Institution		Tasks	Sizes	Runs
TUD	*Tech. Univ. Darmstadt, Dept. of CS, Ubiquitous Knowledge Processing Lab*	DE	Main, EN, DE, FR	S(4), M(4), L(4), XL(4)	16
UniNE	*Univ. Neuchatel - Computer Science*	CH	Main	S(7), XL(1)	8
uscom	*Santiago de Compostela Univ. - Dept. Electronica y Computacion*	ES	Main	S(8)	8
UTASICS	*University of Tampere - Info Studies & Interactive Media and Swedish Institute of Computer Science*	FI SE	Main	XL(8)	8
clefip-ug	*Glasgow Univ. - IR Group Keith*	UK	Main	M(4), XL(1)	5
clefip-unige	*Geneva Univ. - Centre Universitaire d'Informatique*	CH	Main	XL(5)	5
cwi	*Centrum Wiskunde & Informatica - Interactive Information Access*	NL	Main	M(1), XL(4)	4
hcuge	*Geneva Univ. Hospitals - Service of Medical Informatics*	CH	Main, EN, DE, FR	M(3), XL(1)	4
humb	*Humboldt Univ. - Dept. of German Language and Linguistics*	DE	Main, EN, DE, FR	XL(4)	4
clefip-dcu	*Dublin City Univ. - School of Computing*	IR	Main	XL(3)	4
clefip-run	*Radboud Univ. Nijmegen - Centre for Language Studies & Speech Technologies*	NL	Main, EN	S(2)	1
Hildesheim	*Hildesheim Univ. - Information Systems & Machine Learning Lab*	DE	Main	S(1)	1
NLEL	*Technical Univ. Valencia - Natural Language Engineering*	ES	Main	S(1)	1
UAIC	*Al. I. Cuza University of Iasi - Natural Language Processing*	RO	EN	S(1)	1

CLEF–IP 2009. Table 1 contains a list of all submitted runs. Experiments ranged over all proposed tasks (one main task and three language tasks) and over three (S, M, XL) of the proposed task sizes.

To have an overview of the techniques used at CLEF–IP 2009, we looked at how participants approached:

- indexing of the target data
- query generation and ranking of retrieved items

Tables 2 and 3 present at–a–glance a summary of all employed indexing and retrieval methods and, respectively, patent fields used for indexing and query formulation. More details on the participants' approaches are provided in Tables 7 (indexing) and 8 (querying and ranking) in the Appendix.

In Table 2, we marked the usage of some kind of automated translation (MT) in the second column. Methods used for selecting query terms are listed in the third column. As CLEF–IP topics are whole patent documents, many participants found it necessary to apply some kind of term selection in order to limit the number of terms in the query. Methods for term selection based on term weighting are shown here while pre–selection based on patent fields is shown separately in Table 3. Given that each patent document could contain fields in up to three languages, some participants chose to build separate indexes per

Table 2. Index and query formation overview

Group-Id	MT	Term selection	Indexes	Ranking	System
cwi	-	tf–idf	?	boolean, bm25	MonetDB, XQuery, SQL queries
clefip–dcu	-	none	one English only	Indri	Indri
hcuge	x	none	?	bm25	Terrier
Hildesheim	-	none	one English, one German	?	Lucene
humb	x	?	one per language, one phrase index for English, cross–lingual concept index	kl, bm25	PATATRAS
NLEL	-	random walks	mixed language passage index, per year and language	passage similarity	JIRS
clefip–run	-	none	one English only	tf–idf	Lemur
TUD	-	none	one per language, one for IPC	tf–idf	Lucene
UAIC	-	none	one mixed language index (split in 4 indexes for performance reasons)		Lucene
clefip–ug	-	tf–idf	one mixed language	bm25, cosine	IndriLemur
clefip–unige	-	?	one English only	tf–idf, bm25, Fast	?
UniNE	-	tf–idf	one mixed language index	tf–idf, bm25, Dfr	?
uscom	-	tf–idf	one mixed language index	bm25	IndriLemur
UTASICS	x	ratf, tf–idf	1 per language, 1 for IPC	Indri based	IndriLemur

language, while others generated one mixed–language index or used text fields only in one languages discarding information given in the other languages. The granularity of the index varied, too, as some participants chose to concatenate all text fields into one index, while others indexed different fields separately. In addition several special indexes like phrase or passage indexes, concept indexes and IPC indexes were used. A summary on which indexes were built and which ranking models were applied is given in Table 2.

As can be seen in Table 3, the text fields title, claims, abstract and description were used most often. Among the bibliographic fields IPC was the field exploited most, it was used either as post-processing filter or as part of the query. Only two groups used the citation information that was present in the document set. Other very patent-specific information like priority, applicant, inventor information was only rarely used.

As this was the first year for CLEF–IP many participants were absorbed with understanding the data and task and getting the system running. The CLEF–IP track presented several major challenges

- a new retrieval domain (patents) and task (prior art);
- the large size of the collection;
- the special language used in patents (participants had not only to deal with German, English and French text but also with the specialities of patent-specific language);

Table 3. Fields used in indexing and query formulation

		Fields used in index				Fields used in query				
Group-Id	IPC	title	claims	abs	desc	title	claims	abs	desc	Other
cwi	x	x	x	x	x	x	x	x	x	-
clefip–dcu	x	x	x	x	x	x	x	x	x	-
hcuge	x	x	x	x	x	x	x	x	x	citations
Hildesheim	-	x	x	-	-	x	x	-	-	-
humb	x	x	x	x	x	x	x	x	x	citations, priority, applicant, ECLA
NLEL	-	x	-	x	x	x	-	x	-	-
clefip–run	-	-	x	-	-	-	x	-	-	-
TUD	x	x	x	x	x	x	x	-	-	-
UAIC	-	x	x	x	x	x	x	x	x	-
clefip–ug	x	x	x	x	x	x	x	x	x	-
clefip–unige	x	x	x	x	-	x	x	x	x	applicant, inventor
UniNE	x	x	x	x	x	x	x	x	x	-
uscom	x	x	x	x	x	x	x	x	x	-
UTASICS	x	x	x	x	x	x	x	x	x	-

– the large size of topic representations (while in most CLEF tracks a topic consists of few selected query words, for CLEF–IP a topic consists of a whole patent document).

Concerning cross–linguality: not all participants focused on the multilingual nature of the CLEF–IP document collection. In most cases they used only data in one specific language or implemented several monolingual retrieval systems and merged their results. Two groups made use of machine translation: **UTASICS** used Google translate in the Main task to make patent-fields available in all three languages. They report that using the Google translation engine actually deteriorated their results. **hcuge** used Google translate to generate the fields in the missing languages in the monolingual tasks. **humb** applied cross-lingual concept tagging.

Several teams integrated patent–specific know–how in their retrieval systems by using:

classification information IPC and ECLA were found most helpful. Several participants used the IPC class in their query formulation as a post-ranking filter criterium. While using IPC classes to filter out generally improves the retrieval results but it also makes it impossible to retrieve relevant patents that don't share an IPC class with the topic.

citations the citation information included in the patent corpus was exploited by **hcuge** and **humb**.

bibliographic data further bibliographic data such as inventor, applicant, priority information was exploited only by **humb**.

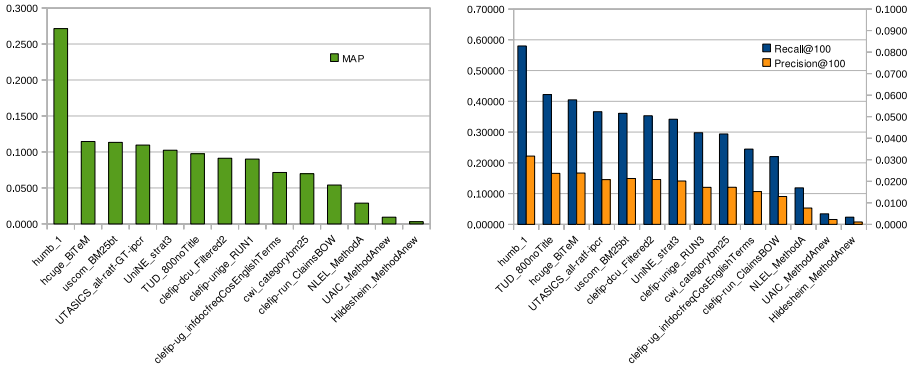


Fig. 3. MAP, Precision@100 and Recall@100 of best run/participant (S)

4 Evaluation

Being aware that the desired result of a patent search is usually given by the combination of several search results, we evaluated the experiments computing some of the most commonly used metrics for IR effectiveness evaluation. The high recall that a prior art search should ultimately deliver is usually obtained through an interactive process that was beyond the scope of this year's investigation.

In addition to the raw measures, we also present here the results of some additional analysis of the results: a correlation analysis for the different topic size bundles showing that the rankings of the systems obtained with different topic sizes can be considered equivalent; a permutation test that shows that not all differences among systems are statistically significant. Furthermore, we started investigating the features that make certain citations harder to find than others with the ultimate goal of being able to make some prediction on the hardness of a given topic. While 13% of the topics contained just title and bibliographic data, we observed that these were not the only topics for which most retrieval methods were unsuccessful. This investigation is ongoing work.

Finally, we mention the results obtained with the manually assessed topics. These assessments obtained from patent experts were meant to provide some clues on the completeness of the automatically generated set of relevance assessments.

4.1 Measurements

The complete collection of measured values for all evaluation bundles is provided in the CLEF–IP 2009 Evaluation Summary ([13]). Detailed tables for the manually assessed patents are provided in a separate report ([14]).

After minor data format corrections, we created experiment bundles based on size and task. For each experiment we computed 10 standard IR measures:

- Precision, Precision@5, Precision@10, Precision@100
- Recall, Recall@5, Recall@10, Recall@100

Table 4. MAP, Precision@100, Recall@100 of best run/participant (S)

Group–Id	Run–Id	MAP	Recall@100	Precision@100
humb	1	0.2714	0.57996	0.0317
hcuge	BiTeM	0.1145	0.40479	0.0238
uscom	BM25bt	0.1133	0.36100	0.0213
UTASICS	all–ratf–ipcr	0.1096	0.36626	0.0208
UniNE	strat3	0.1024	0.34182	0.0201
TUD	800noTitle	0.0975	0.42202	0.0237
clefip–dcu	Filtered2	0.0913	0.35309	0.0208
clefip–unige	RUN3	0.0900	0.29790	0.0172
clefip–ug	infdocfreqCosEnglishTerms	0.0715	0.24470	0.0152
cwi	categorybm25	0.0697	0.29386	0.0172
clefip–run	ClaimsBOW	0.0540	0.22015	0.0129
NLEL	MethodA	0.0289	0.11866	0.0076
UAIC	MethodAnew	0.0094	0.03420	0.0023
Hildesheim	MethodAnew	0.0031	0.02340	0.0011

– MAP
– nDCG (with a reduction factor given by a logarithm in base 10).

All computations were done with SOIRE, a software for IR evaluation based on a service–oriented architecture ([4]). Results were double–checked against `trec_eval`[11], the standard program for evaluation used in the TREC evaluation campaign, except for nDCG for which, at the time of the evaluation, no implementation was publicly available. We note, here, that the nDCG version implemented in SOIRE did not take into account a cumulated gain, as described in [10]. We believe that the nDCG version we used considers that the user will always be happy to get a relevant document, but it will be more tired, the more documents she has to look at.

MAP, recall@100 and precision@100 of the best run for each participant are listed in Table 4 and illustrated in Figure 3. The values shown are those computed for the small topic set. The MAP values range from 0.0031 to 0.27 and are quite low in comparison with other CLEF tracks. There are three considerations that have to be made concerning these low precision values.

Firstly, it must be noted that the average topic had 6 relevant documents, meaning that the upper boundary for precision@100 was at 0.06.

Furthermore, these low values are in part due to the incompleteness of the automatically generated set of relevance assessments: some of the target citations were in fact almost impossible to find since they contained no textual fields except for the patent's title or just title and abstract.

Last but not least, the low values reflect the difficulty of the patent retrieval task.

[11] http://trec.nist.gov/trec_eval

4.2 Analysis of Results

Correlation analysis In order to see whether the evaluations obtained with the three different bundle sizes (S, M, XL) could be considered equivalent we did a correlation analysis comparing the vectors of MAPs computed for each of the bundles.

Table 5. Correlations of systems rankings for MAP

Correlation	#runs	τ	ρ
M vs XL	24	0.9203	0.9977
S vs M	29	0.9160	0.9970
S vs XL	24	0.9058	0.9947
XL vs ManXL	24	0.5	0.7760
M vs ManM	29	0.6228	0.8622
S vs ManS	48	0.4066	0.7031

In addition to that, we also evaluated the results obtained by the track's participants for the 12 patents that were manually assessed by patent experts. We evaluated the runs from three bundles extracting only the 12 patents (when present) from each runfile. We called these three extra-small evaluation bundles and named them ManS, ManM, ManXL. Table 5 lists Kendall's τ and Spearman's ρ values for all compared rankings.

Figures 4 and 5 illustrate the correlation between pairs of bundles together with the best least–squares linear fit.

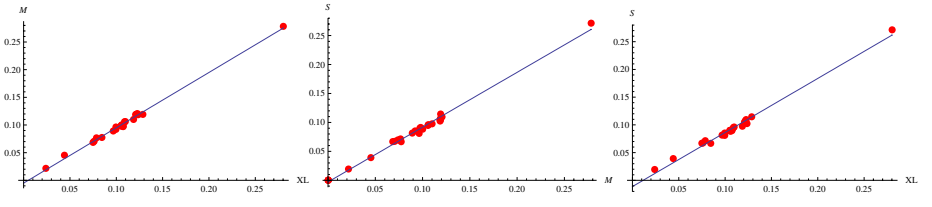


Fig. 4. Correlation of rankings by MAP: comparison of XL, M, S bundles

The rankings obtained with topic sets S, M, and L are highly correlated, suggesting that the three bundles an be considered equivalent for evaluation purposes. As expected, the correlation between S, M, XL and the respective ManS, ManM, ManXL rankings by MAP drops drastically.

It must however be noted that the limited number of patents in the manual assessment bundle (12) is not sufficient for drawing any conclusion. We hope to be able to collect more data in the future in order to assess the quality of our automatically generated test collection.

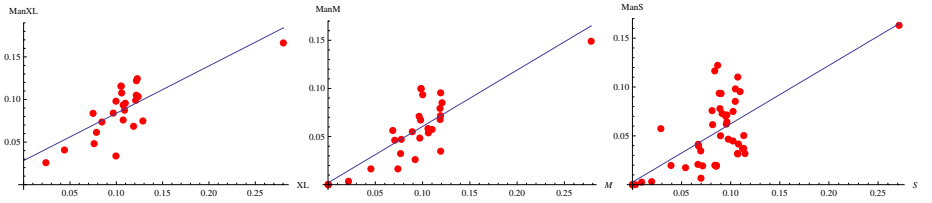


Fig. 5. Correlation of rankings by MAP: bundles XL, M, S versus manual relevance assessments ($\leq$ 12 topics)

Some remarks on the manually assessed patents. Patent experts marked in average 8 of the proposed patents as relevant to the topic patent. For a comparison:

- 5.4 is the average number of citations for the 12 topic patents that were assessed manually;
- for the whole collection, there are in average 6 citations per patent.

Furthermore, some of the automatically extracted citations (13 out of 34) were marked as not relevant by patent experts. In order to have some meaningful results a larger set of data and an agreement on the concept of relevance are needed.

Statistical significance of evaluation results. In order to determine the statistical significance of our evaluation results we ran a Fisher's permutation (or randomization) test as proposed in [16]. We ran the test only on the small bundle of runs. This additional analysis allowed us to detect those differences that are statistically significant with a 95% confidence ($p = 0.05$).

In Table 6 MAPs of the best run for each participant are listed and whenever no significant difference was observed the columns are grouped together. Note that by increasing the significance level a bit - for instance for $p = 0.1$ - the second grouping from the left (UTASICS–UniNe) would disappear.

Table 6. Runs for which no statistically significant difference was detected ($p \geq 0.05$)

Group–Id	Run–Id	p	Group–Id	Run–Id
clefip–unige	RUN1	0.17275	TUD	800noTitle
hcuge	BiTeM	0.85141	uscom	BM25bt
hcuge	BiTeM	0.40999	UTASICS	all-ratf-GT-ipcr
TUD	800noTitle	0.41601	UniNE	strat3
UniNE	strat3	0.07783	UTASICS	all-ratf-GT-ipcr
uscom	BM25bt	0.4258	UTASICS	all-ratf-GT-ipcr

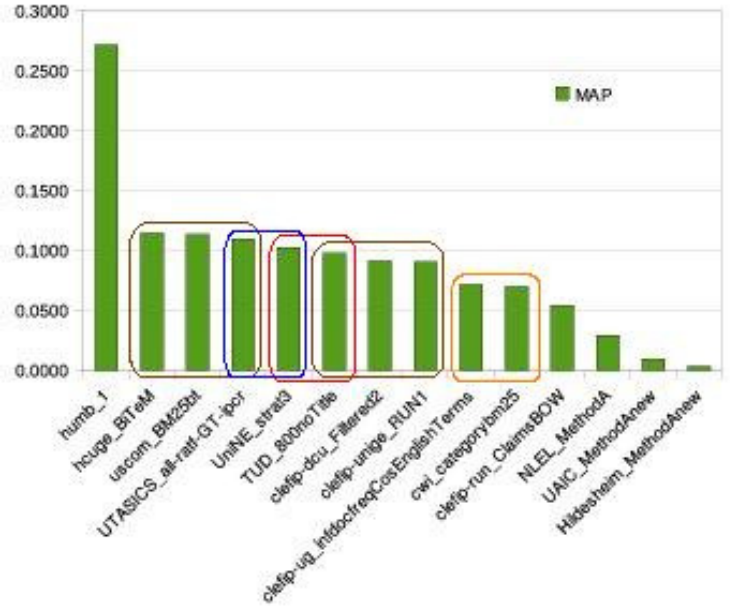


Fig. 6. MAP, Precision@100 and Recall@100 of best run/participant (S)

5 Lessons Learned and Plans for 2010

The patent search task was a brand new challenge for both organizers and track participants. All beginnings are difficult, but we believe that we made a good step forward in understanding the intricacies of patent retrieval and in learning how to adapt Information Retrieval to tackle this challenge. The communication with patent professionals was difficult in the beginning but it improved over time; their contribution is essential for ensuring the plausibility of the CLEF–IP evaluation track.

The prior art search task must target patent applications as sources for topics, since patent citations refer to application documents. CLEF–IP 2009 used granted patents as the main topic source with the intent to have a topic set with parallel, multilingual content. As it turned out, from an IP expert's point of view, patent applications and granted patents differ significantly when confronted with the patent application's citations.

In the evaluation, we included family members in the set of extended citations. While these can be considered prior art items, they do not really improve retrieval effectiveness because, for a patent searcher, members of the same patent family are equivalent. This issue should be considered when evaluating results by measuring one single hit per family.

One interesting extension of the patent citations methodology would be to consider forward citations for finding relevant documents. This is apparently a commonly adopted method among patent search professionals. The resulting set of relevance assessments would clearly contain items of lower relevance and it would be interesting to see what is the trade-off (more relevant versus more irrelevant patent documents) of such an enlarged set of relevance assessments.

In the 2009 collection only patent documents with data in French, English and German were included. One area in which to extend the track would be to provide additional patent data in more European languages.

Clef–Ip chose topics at random from the topics pool, thus topics have the same distribution of languages as in the data pool (70%, 23%, 7% documents respectively in English, German, French). In order to compare language-specific retrieval techniques, an even distribution of topics among languages would allow a fairer comparison.

Patents are organized in what are known as "patent families". A patent might be originally filed in France in French, and then subsequently to ease enforcement of that patent in the United States a related patent might be filed in English with the US Patents and Trademarks Office. Although the full text of the patent will not be a direct translation of the French (for example because of different formulaic legal wordings) the two documents may be comparable, in the sense of a Comparable Corpus in Machine Translation). It might be that such comparable data will be useful to participants to mine for technical and other terms. The 2009 collection does not lend itself to this use and we will seek to make the collection more suitable for that purpose.

For the first year we measured the overall effectiveness of systems. A more realistic evaluation should be layered in order to measure the contribution of each single component to the overall effectiveness results as proposed in the GRID@CLEF track ([5]) and also by [8].

The 2009 task was also somewhat unrealistic in terms of a model of the work of patent professionals. It is unclear how well the 2009 task and methodology maps to what makes a good (or better) system from the point of view of patent searchers. Real patent searching often involves many cycles of query reformulation and results review, rather than one off queries and results set. A more realistic model that also includes the interactive aspect of patent search might be the subject of future evaluations.

6 Epilogue

Clef–Ip has to be regarded as a major success: looking at previous Clef tracks we regarded four to six groups as a satisfactory first year participation rate. Fifteen is a very satisfactory number of participants—a tribute to those who did the work and to the timeliness of the task and data. In terms of retrieval effectiveness the results have proved hard to evaluate: if there is to make an overall conclusion, then the effective combination of a wide range of indexing methods is best, rather than a single silver bullet or wooden cross. Still, some of

the results from groups other than Humboldt University indicate that specific techniques may work well.

Finally we need to be clear that a degree of caution is needed for what is inevitably an initial analysis of a very complex set of results.

Acknowledgements

We thank Judy Hickey, Henk Tomas and all the other patent experts who helped us with manual assessments and who shared their know–how on prior art searches with us.

References

1. European Patent Convention (EPC), http://www.epo.org/patents/law/legal-texts
2. Guidelines for Examination in the European Patent Office (2009), http://www.epo.org/patents/law/legal-texts/guidelines.html
3. Adams, S.R.: Information sources in patents. K.G. Saur (2006)
4. Dittenbach, M., Pflugfelder, B., Pesenhofer, A., Roda, G., Berger, H.: Soire: A Service-Oriented IR Evaluation Architecture. In: CIKM 2009: Proceeding of the 18th ACM Conference on Information and Knowledge Management, pp. 2101–2102. ACM, New York (2009)
5. Ferro, N., Harman, D.: Dealing with multilingual information access: Grid experiments at trebleclef. In: Agosti, M., Thanos, C. (eds.) Post-proceedings of the Fourth Italian Research Conference on Digital Library Systems, IRCDL 2008 (2008)
6. Fujii, A., Iwayama, M., Kando, N.: Overview of the Patent Retrieval Task at the NTCIR-6 Workshop. In: Kando, N., Evans, D.K. (eds.) Proceedings of the Sixth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan, pp. 359–365. National Institute of Informatics (2007)
7. Graf, E., Azzopardi, L.: A Methodology for Building a Patent Test Collection for Prior art Search. In: Proceedings of the Second International Workshop on Evaluating Information Access, EVIA (2008)
8. Hanbury, A., Müller, H.: Toward automated component–level evaluation. In: SIGIR Workshop on the Future of IR Evaluation, Boston, USA, pp. 29–30 (2009)
9. Hunt, D., Nguyen, L., Rodgers, M.: Patent searching: tools and techniques. Wiley, Chichester (2007)
10. Järvelin, K., Kekäläinen, J.: Cumulated gain–based evaluation of IR techniques. ACM Trans. Inf. Syst. 20(4), 422–446 (2002)
11. Kando, N., Leong, M.-K.: Workshop on Patent Retrieval (SIGIR 2000 Workshop Report). SIGIR Forum 34(1), 28–30 (2000)
12. Organisation for Economic Co–operation and Development (OECD). OECD Patent Statistics Manual (Feburary 2009)
13. Piroi, F., Roda, G., Zenz, V.: CLEF-IP 2009 Evaluation Summary (July 2009)
14. Piroi, F., Roda, G., Zenz, V.: CLEF-IP 2009 Evaluation Summary part II (September 2009)
15. Roda, G., Zenz, V., Lupu, M., Järvelin, K., Sanderson, M., Womser-Hacker, C.: So Many Topics, So Little Time. SIGIR Forum 43(1), 16–21 (2009)

16. Smucker, M.D., Allan, J., Carterette, B.: A comparison of statistical significance tests for information retrieval evaluation. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, pp. 623–632. ACM, New York (2007)
17. Tiwana, S., Horowitz, E.: Findcite – automatically finding prior art patents. In: PaIR 2009: Proceeding of the 1st ACM Workshop on Patent Information Retrieval. ACM, New York (to appear)
18. Voorhees, E.M.: Topic set size redux. In: SIGIR 2009: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 806–807. ACM, New York (2009)
19. Voorhees, E.M., Buckley, C.: The effect of topic set size on retrieval experiment error. In: SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 316–323. ACM, New York (2002)

Appendix

Table 7. Indexing the data

Group ID	IPC	title	clms	abs	desc	System	Other notes
cwi	x	x	x	x	x	MonetDB/XQuery PF/Tijah module	– domain specific index containing relations (e.g. inventor and patents, IPC and docs) – domain unaware index – stored in a SQL database – split on 4 different databases
clefip–dcu	x	x	x	x	x	Indri	– merged patent docs into one doc, using last versions of fields in docs. – English only – where fields were missing, content in other fields was used to fill them up (abs into desc, or claims; title into abs, desc, claims, etc.) – list of stop words extracted by term freq in fields across all docs
hcuge	x	x	x	x	x	Terrier	IPC once limited to 4 char, once complete
Hildesheim	-	x	x	-	-	Lucene	– one English and one German index – German Analyser on German content
humb	x	x	x	x	x	PATATRAS with – MySQL to store metadata – own HMM–based implementation for English content – Tree Tagger for French and German	– further metadata used included patent applicants' name and address, citations – indexes built at lemma level; for English an additional phrase index; concept tagging – crosslingual concept index (multilingual terminological database)
NLEL	-	x	-	x	x	JIRS (for Passage Retrieval) developed at Univ. Poli. de Valencia	– used only one doc from a set of patent documents (the 'last' version/kind) – 1 per year and language
clefip–run	-	-	x	-	-	Indri/Lemur	used a stop word list for English
TUD	x	x	x	x	x	– preprocessing: UIMA (Unstruc. Infor. Management Archit.)in DKPro Infor Retr framework; – indexing: Lucene	– used one doc per patent, with fields available in the 'latest' doc kinds – preprocess: sentence splitting, tokenization, stopwords removal, stemming, compound splitting – one index per language – topic patents were also indexed separately (title & claims only)
UAIC	-	x	x	x	x	adapted Lucene indexer	– desc is used/indexed only if abs is not available – 4 parallel indexes (peer–to–peer) – English only
clefip–ug	x	x	x	x	x	Indri/Lemur	Did not index the separate fields, but the doc as a whole, with a minimal list of stopwords
clefip–unige	x	x	x	x	-	?	– English only – used applicant and inventor fields, too
UniNE	x	x	x	x	x	?	– used stemmers and stopwords lists for each language; – one mixed language index
uscom	x	x	x	x	x	see clefip–ug	– used one multilingual index of clefip–ug; – indexed all patent docs
UTASICS	x	x	x	x	x	Indri/Lemur	– one index per language; – one IPC index truncated at 4 chars; – indexed a virtual patent (with fields taken from the latest patent doc possible)

Table 8. Queries and retrieves

Group ID	title	clms	abs	desc	IPC	Query generation method	Retrieval system	Ranking results	Tranlsate	Other notes
cwi	x	x	x	x	x	– visual interface, drag'n'drop to create complex search strategies – translated into PRA, then SQL queries – term selection: tf–idf	MonetDB SQL queries	– bm25, boolean – category (IPC based)	–	HySpirit software component for PRA[12] is developed by Apriorie
clefip–dcu	x	x	x	x	x	– topic docs processed as those for indexing – abs and desc used only for English topics – bi–gram words	Indri	Indri	–	IPC classes used to filter out results
hcuge	x	x	x	x	x	Terrier 2.2.1	Terrier 2.2.1	– Terrier – bm25, tf–idf, bb2, etc.	x[13]	Citations and IPCs used to post–process results
Hildesheim	x	x	-	-	-	Simple term queries	Lucene	Lucene based	–	Later experiments used IPCs and Snowball stemmer.
humb	x	x	x	x	x	Topic docs processed as docs for indexing	PATATRAS – Lemur toolkit – Unigram, KL, Okapi	– result lists merged using SVM regression models and liniar comb of normalized ranking scores – post–ranking using SVM regression model	x[14]	– reduced topic search space by using working sets per topic. – citations, applicant address, name, ECLA codes
NLEL	x	-	x	-	-	Summarization by random walks, from the topic original language (only claims were for sure in all lang)	Adapted JIRS for Passage Retrieval	JIRS based	x[15]	One query per language
clefip–run	-	x	-	-	-	Complete claims	Lemur	tf–idf	–	
TUD	x	x	-	-	x	– IPC as a separate query – one query per lang – min 800 words (from title, clms and abs or desc when claims field was shorter)	Lucene based	Lucene based (tf idf?)	–	
UAIC	x	x	x	x	-	Lucene based	Lucene	Lucene based, with boost factors		desc is used/indexed only if abs is not available;

Continued on next page

[12] Probabilistic Relational Algebra
[13] Google Translate for the language tasks where fields were missing
[14] Multilanguage terminology database
[15] Google Translate

Table 8. *continued*

Group ID	title	clms	abs	desc	IPC	Query generation method	Retrieval system	Ranking results	Tranlsate	Other notes
clefip–ug	x	x	x	x	x	– df, tf, query length thresholds; – 1: computes distribution of terms to related patents – 2: based on this distribution, does query extraction from the topics	Indri/Lemur	bm25	–	
clefip–unige	x	x	x	x	x	whole patent?	?	tf–idf, Okapi, Fast	–	– filtering of results by IPC class – filtering on the length of docs
UniNE	x	x	x	x	x	tf–idf; max 100 terms	?	– tf–idf with cosine normalization; – BM25, DFR (divergence from randomness) – LM (stat language model)	–	no workshop notes
uscom	x	x	x	x	x	– idf – extracted query terms per language, then compiled them – no. of query terms: a) fixed; b) percentage of topic length	bm25 (Lemur)	– given by bm25 – the patent doc highest in result list was kept	–	
UTASICS	x	x	x	x	x	– ratf term selection before Google Translate – tf–idf – max 50 terms	Indri (Lemur)	– rank given by Indri on the 4 queries and – Mean Average Distance to combine query results (involves relative weight computation)	x[16]	– 3 queries (one per lang) + one IPC query – various combinations of fields (abs + desc, abs + claims, etc) – additional manual queries with up to 10 keywords (IP experts involved)

[16] Google Translate to get query terms where fields missed content in the query language

Exploring Structured Documents and Query Formulation Techniques for Patent Retrieval

Walid Magdy, Johannes Leveling, and Gareth J.F. Jones

Centre for Next Generation Localization
School of Computing
Dublin City University, Dublin 9, Ireland
{wmagdy,jleveling,gjones}@computing.dcu.ie

Abstract. This paper presents the experiments and results of DCU in CLEF-IP 2009. Our work applied standard information retrieval (IR) techniques to patent search. Different experiments tested various methods for the patent retrieval, including query formulation, structured index, weighted fields, document filtering, and blind relevance feedback. Some methods did not show expected good retrieval effectiveness such as blind relevance feedback, other experiments showed acceptable performance. Query formulation was the key to achieving better retrieval effectiveness, and this was performed through assigning higher weights to certain document fields. Further experiments showed that for longer queries, better results are achieved but at the expense of additional computations. For the best runs, the retrieval effectiveness is still lower than for IR applications for other domains, illustrating the difficulty of patent search. The official results have shown that among fifteen participants we achieved the seventh and the fourth ranks from the mean average precision (MAP) and recall point of view, respectively.

1 Introduction

This paper presents the experimental results of Dublin City University (DCU) in the CLEF-IP track 2009. We participated in the main task which is retrieving patents prior art. The aim of the task is to automatically retrieve all of citations for a given patent (which is considered as the topic) [5]. Only three runs were submitted, but additional unofficial experiments were performed for this task. Fifteen participants have submitted 48 runs; according to MAP scores, our best run achieved the seventh rank across participants and the 22nd across all runs. However, according to recall scores, our best run achieved the fourth rank across all participants and the fourth rank across all 48 submitted runs.

The paper is organized as follows: Section 2 describes the data for the task and an analysis of its nature; Section 3 presents all the experiments for this task; Section 4 shows the results; then Section 5 discusses these results; Finally, Section 6 concludes the paper and provides possible future directions.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 410–417, 2010.

2 Data Pre-processing

More than 1.9M XML documents were provided representing different versions of 1M patents filed between 1985 and 2000. For our experiments, all different document versions for a single patent were merged into one document with fields updated from its latest versions. Patent structure is very rich, and some fields are present in three languages (English "EN", German "DE", and French "FR"), namely the title and claims. Only the patent 'title', 'abstract', 'description', 'claims', and 'classifications' fields are extracted from the patents. However, many patents lack some of these fields. The only fields that are present in all patents are the title and the classifications; the other fields are omitted in some patents. The "description" field is related to the "claims" field, and if the "claims" field is missing, then "description" is missing too. However, the opposite in not true, as some documents contain a "claims" field while the "description" field is missing. The "abstract" field is an optional part that is present in some patents. About 23% of the patents do not contain the claims and description fields, out of which 73% only have titles. 54% of the patents have claims in three languages (English, French, and German), and the remainder 23% of the patents have claims in the document language only (language of the 'description' field), these 23% are 68% English, 23% German, and 9% French.

In order to avoid language problems, the English fields only are selected. This step will lead to the loss of extra 7.4% of the patents which lack the claims and description fields (these are the German and French patents with claims only in one language). In addition, all non-English patents lack the abstract and description fields. The final outcome resulted in 30% of the collection suffering from missing most of the fields. This portion of the collection mostly comprises the titles only with a small portion of it containing abstracts too.

In order to maintain the full structure and overcome the lack of some fields in some patents, the abstract (if it exists) is copied to the description and claims fields; otherwise, the title is used instead.

3 Experimental Setup

In this section, different experiments for indexing and searching the data are discussed. After merging different versions of patents and extracting the relevant fields, some pre-processing is performed for the patent text in order to prepare it for indexing. Different methods were used for query formulation to search the collection.

Many experiments were performed on the training topics provided by the task organizers, however, a small number was submitted on the test data for the official runs. The training set contains 500 patent topics, which was sufficient to compare different methods and select the best for the official submissions. Official experiments were performed on the X-large topics set consisting of 10,000 patent topics. For each topic, the top 1,000 documents are retrieved.

3.1 Text Pre-processing

Patent text contains many formulas, numeric references, chemical symbols, and patent-specific words (such as *method*, *system*, or *device*) that can cause a negative effect

on the retrieval process. Some filtering of the text is done by removing predefined stop words[1], digits, and field-specific stop words.

To obtain the fields stop words, the field frequency for terms is calculated separately for each field. The field frequency for a term "T" in field "X" is the number of fields of type "X" across all documents containing the term "T". For each field, all terms with field frequency higher than 5% of the highest term field frequency for this field are considered as stop words. For example, for the "title" field, the following words have been identified as stop words: *method*, *device*, *apparatus*, *process*, etc; for another field such as "claims", the following words have been identified as stop words: *claim*, *according*, *wherein*, *said*, etc.

3.2 Structured Indexing

Indri [6] was used to create a structured index for patents. A structured index keeps the field structure in the index (Figure 1). This structured index allows searching specific fields instead of searching in the full document. It also allows giving different weights for each field while searching. As shown in Figure 1, "DESC1" and "CLAIM1" are sub-fields for the description "DESC" and claims "CLAIMS" fields respectively. "DESC1" is the first paragraph in the description field; typically it carries useful information about the field of the invention and what the invention is about. "CLAIM1" is the first claim in the claims sections, and it describes the main idea of the invention in the patent. The field "CLASS" carries the IPC classification [7] information of the patent of which the three top classification levels are used, the deeper levels are discarded (example: B01J, C01G, C22B).

As mentioned earlier, for patents that lack some fields, the empty fields are filled with the abstract if it exists or with the title otherwise. Pre-processing includes stemming using the Porter stemmer [4].

```
<DOC>
      <DOCNO>patent number</DOCNO>
      <TEXT>
            <TITLE>title</TITLE>
            <CLASS>3rd level classification</CLASS>
            <ABSTRACT>abstract</ABSTRACT>
            <DESC>
            <DESC1>1st sentence in description</DESC1>
            Rest of patent description
            </DESC>
            <CLAIMS>
            <CLAIM1>1st claim</CLAIM1>
            Rest of patent claims
            <CLAIMS>
        </TEXT>
</DOC>
```

Fig. 1. Structured text for a patent in TREC format

3.3 Query Formulation

Query formulation can be seen as one major task in patent retrieval ([1], [5]). As a full patent is considered to be the topic, extracting the best representative text with the proper weights is the key enabling for good retrieval results.

[1] http://members.unine.ch/jacques.savoy/clef/index.html

Using the full patent as a query is not practical due to the huge amount of text in one patent. Hence, text from certain fields was extracted and tested to search the structured index with different weights to different fields. Various combinations of fields were employed, using different weights, enabling/disabling filtering using third level classification, and enabling/disabling blind relevance feedback [6].

The patent topic text was pre-processed in the same way as in the indexing phase by removing stop words and digits, in addition to removing special characters, symbols and all words of small length (one or two letters).

Similar to the indexed documents, only English parts are used, which means all non-English patent topics will miss the abstract and description fields to be used in the search. However, the amount of text present in claims and titles should be sufficient to create a representative query. In patent topics, claims and titles are always present in all three languages. Two types of experiments for the query formulation were conducted, the first type focused on using the short text fields to create the query from the patent topic. The short text fields that were used for constructing the queries are "title", "abstract", "desc1" (first line in description), "claim_main" (first sentence in first claim), "claim1" (first claim), and "claims". The second type of experiments tested using the full patent description as the query, which does not exist for non-English patent topics; hence, the already existing translated parts of the non-English patents are used instead.

The aims behind both types of experiments are to check the most valuable parts that better represent the patent, and to check the possibility of reducing the amount of query text which leads to less processing time without reducing the quality of results.

3.4 Citations Extraction

One of the strange things about patents, and that is thought to be neglected or forgotten by the track organizers, is the presence of some of the cited patents numbers within the text of the description of the patents. These patent numbers have not been filtered out of the text of the patent topics, which can be considered as the presence of part of the answer within the question. Despite of this fact, we have not focused on building extra experiments based on this information as it can be considered as a hack for finding the cited patents. In addition, in real life this information is not always presented in the patent application, and hence, creating results on it can be considered as a misleading conclusion in the area of patent retrieval.

However, in the results, adding this information to the tested methods is reported to demonstrate the impact of using this kind of information. Results shows that a misleading high MAP can be achieved but with a very low recall, and recall is usually the main objective for the patent retrieval task.

For the X-large topics collection which contains 10,000 patent topics, 36,742 patent citations were extracted from the patent topics, but only 11,834 patents citations were found to be in the patent collection. The 11,834 patent citations are extracted from 5,873 patent topics, leaving 4,127 topics with nothing extracted from them. Only 6,301 citations were found to be relevant leading to a MAP of 0.182 and a recall of 0.2 of the cited patents to these topics. The format of the cited patent number within the description text varies a lot; hence, we think that more cited patents could be extracted from the text if more patterns for the citations were known to us.

4 Submitted Runs and Results

Some of the tested methods seemed to be ineffective for our IR experiments. Blind relevance feedback (FB) and structured search have negative impact on the results (best FB run achieved 0.05 MAP) . All experiments with blind relevance feedback led to a degradation in the MAP to around 60% of the original runs without feedback, and this can stem from the low quality of the highly ranked results. Structured retrieval was tested by searching each field in the patent topic to its corresponding field in the index. Different weights for fields were tested; however, all experiments led to lower MAP and recall than searching in the full index as a whole without directing each field to its correspondent. Since patent documents were treated as full documents neglecting their structure, patent topics which were used for formulating the queries were tested by giving different weights to the text in each short field and compared to using the full description for formulating the query. Assigning higher weight to text in "title", "desc1", and "claim_main" has been proven to produce the best results across all runs for using the short fields.

Three runs were submitted to CLEF-IP 2009 on the official topics with the same setup which returned the best results in training. The three runs tested how to better use the short fields to generate the query. The common setup for the three runs was as follows:

1. The patent document is treated as a full document, neglecting its structure.
2. English text only is indexed with stemming (Porter stemmer).
3. Stop words are removed, in addition to digits and words consisting of less than two letters.
4. A query is formulated from the following fields with the following weights: 5×title + 1×abstract (English topics only) + 3×desc1 (English topics only) + 2×claim_main + 1×claims.
5. Additional bi-grams with a frequency in the text higher than one were used in query. The text of the fields: "title", "abstract", "desc1", and "claim_main" was used for extracting the bi-grams words.

The difference between the three runs is as follows:

- Run 1: No filtering to the results is performed.
- Run 2: Filtering is performed for all results that do not match up to the third level classification code of the patent topic (at least one common classification should be present).
- Run 3: The same as 2nd run, but removing query words consisting of less than three letters.

Runs were submitted on the X-large topic collection that contains 10,000 patent topics. The average time for running this amount of topics was around 30 hours (about ten seconds on average for retrieving results of one topic on a standard 2GB RAM, Core2Duo 1.8GHz PC).

Later experiments tested the use of the full description text of a patent topic to generate the query after removing all terms appeared only once. The average amount of time taken to search one topic was found to be slightly higher than 1 minute, which is more than 6 times the average time taken for searching using the short fields.

Table 1 shows the results of the 3 submitted runs [5]. In Table 1, it is shown that the 3rd run got the best results from the precision and recall perspective. The 1st run yields the lowest performance, which shows that applying the filtering over the results based on the patent classification codes is useful. For all runs (official and training ones), the retrieval effectiveness is relatively low when compared to other IR tasks; this can stem from the nature of patent document itself in addition to the task of finding cited patents which are relevant to the patent topic from the conceptual point of view, not from the word matching. This is discussed in the next section in detail.

In Table 2, the additional experiments when using the description of the patent topics to search the collection are compared to the best run in Table 1 (Run 3). In addition, adding the extracted citations from the description to both results is reported. From Table 2, it can be seen that using the description text for searching is on average 11% better than using the best combination of the short fields from the precision perspective, and this was statistically better when tested using Wilcoxon statistical significance test with confidence level of 95% [3]. Furthermore, combining the results with the extracted citations from the text leads to a huge improvement in the MAP, where these citations are considered as the top ranked results in the final list then added the results from the searching. When combining both results, if more than one citation is extracted from the text on one topic, they are ordered according to their position in the search result list, otherwise, the extracted citations are ordered randomly in the top of the list. Although the impact of adding the extracted citations to the results list is high, it can not be considered as an information retrieval result, as no search effort is done for retrieving these documents, and building a conclusive method for searching patents can not be generalized based on these results as it is not the common case for most of the cited patents.

Table 1. Recall (R) and MAP for the 3 submitted runs in CLEF-IP 2009

Run #	R	MAP
Run 1	0.544	0.097
Run 2	0.624	0.107
Run 3	0.627	0.107

Table 2. Recall (R) and MAP for the best submitted run compared to using patent topic description for search with and without adding extracted citations

	Official Results		With Extracted Citations	
	R	**MAP**	**R**	**MAP**
Run 3	0.627	0.107	0.660	0.200
Description	0.627	0.119	0.668	0.209

5 Discussion

In this section, results are analyzed to identify the reasons behind the low retrieval effectiveness for the patent retrieval task. In order to analyze this problem, the overlap between short fields of each topic in the training data and its relevant cited patents is computed; in addition, the overlap between short fields of the topics and the top five

ranked non-relevant documents is calculated. The reason behind selecting the number "five" is that the average number of relevant documents for all topics is between five and six. The overlap is measured using two measures: 1) cosine measure between each two corresponding fields of the two compared patents; 2) percentage of zero overlap (no shared words) between two corresponding fields of the two compared patents. The same pre-processing is done for all patents and topics, where stop words are removed (including digits), and the comparison is based on the stemmed version of words. From Figure 2 and 3, it seems that relying on common words between topics and relevant documents for patent retrieval is not the best approach. Figure 3 shows that the cosine measure between the top ranked non-relevant documents to the topic is nearly twice as high as for the relevant documents for all fields. The same is shown in Figure 4, where surprisingly, 12% of the relevant documents for topics have no shared words in any field with the topics. This outcome has proven the importance of introducing different approaches for query formulation instead of relying on word matching in the patent topics only.

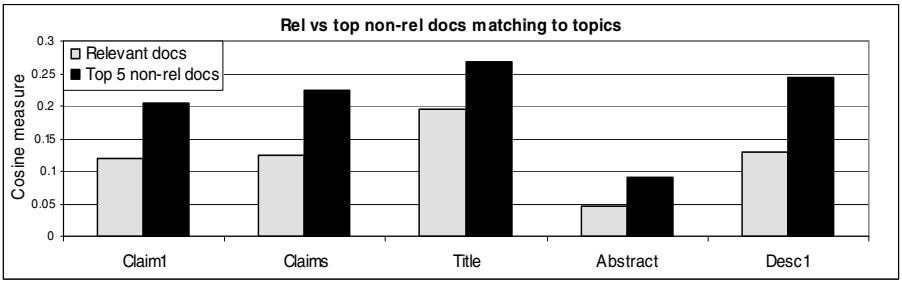


Fig. 2. Cosine measure between fields of topics and the corresponding ones in relevant and top retrieved documents

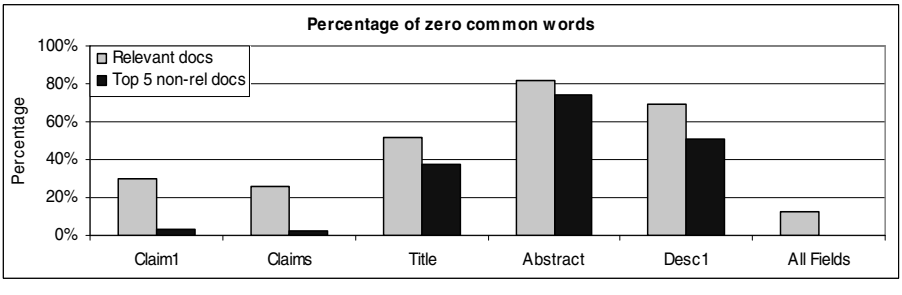


Fig. 3. Percentage of fields with zero common (shared) words between that of topics and the corresponding ones in relevant and top retrieved documents

6 Conclusion and Future Work

In this paper, we described our participation in the CLEF-IP track 2009. Standard IR techniques were tested focusing mainly on query formulation. Our experiments

illustrated the challenge of the patent search task, where an additional analysis showed that depending on word matching is not the best solution as in other IR applications. Our best result was obtained by treating patents as a full document with some pre-processing by removing standard stop words in addition to patent-specific stop words. In the query phase, it was shown that the more text is present in the query the better the results are. However, the computational cost is much higher. For using the short fields for query formulation, text is extracted from these fields and higher weights are assigned to some fields. When using the full patent description text, 11% improvement in the retrieval is achieved, but 6 times the processing time is required. Some additional experiments showed the poor effectiveness of using blind relevance feedback or using the patent structure in index.

For future work, more investigation is required for checking the best use of patent structure in both index and query phases. Machine learning can be a useful approach for identifying the best weights for different fields. Furthermore, query expansion through the conceptual meaning of words is a potential approach to be tested. Finally, machine translation can be a good solution to overcome the problem of multi-lingual documents and queries.

Acknowledgment

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (CNGL) project.

References

[1] Fujii, A., Iwayama, M., Kando, N.: Overview of patent retrieval task at NTCIR-4. In: Proceedings of the Fourth NTCIR Workshop on Evaluation of Information Retrieval, Automatic Text Summarization and Question Answering, Tokyo, Japan, June 2-4 (2004)

[2] Graf, E., Azzopardi, L.: A methodology for building a patent test collection for prior art search. In: EVIA-2008 Workshop, NTCIR-7 (2008)

[3] Hull, D.: Using statistical testing in the evaluation of retrieval experiments. In: SIGIR 1993, New York, NY, USA, pp. 329–338 (1993)

[4] Porter, M.F.: An Algorithm for Suffix Stripping. Program 14(3), 130–137 (1980)

[5] Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: retrieval experiments in the Intellectual Property domain. In: CLEF Working Notes 2009, Corfu, Greece (2009)

[6] Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proceedings of the International Conference on Intelligence Analysis (2004)

[7] IPC (International Patent Classification), http://www.epo.org/patents/patent-information/ipc-reform.html

Formulating Good Queries for Prior Art Search

José Carlos Toucedo and David E. Losada

Grupo de Sistemas Inteligentes
Dept. Electrónica y Computación
Universidad de Santiago de Compostela, Spain
{josecarlos.toucedo,david.losada}@usc.es

Abstract. In this paper we describe our participation in CLEF-IP 2009 (prior art search task). This was the first year of the task and we focused on how to build effectively a prior art query from a patent. Basically, we implemented simple strategies to extract terms from some textual fields of the patent documents and gave more weight to title terms. We ran experiments with the well-known BM25 model. Although we paid little attention to language-dependent issues, our performance was usually among the top 3 groups participating in the task.

1 Introduction

The main task of the CLEF-IP09 track is to investigate information retrieval techniques for patent retrieval, specifically for *prior art search*. Prior art search, which consists of retrieving any prior record with identical or similar contents to a given patent, is the most common type of retrieval in the patent domain.

The track provides the participants with a huge collection of more than one million patents from the European Patent Office (EPO). Every patent in the collection consists of several XML documents (generated at different stages of the patent's life-cycle) that can be written in French, English or German[1].

In an information retrieval setting the patent to be evaluated can be regarded as the information need and all the patent documents (granted patents and applications) filed prior to the topic application as the document collection. However, the query patent provided is a very long document which contains many ambiguous and vague terms [1]. Therefore, this year our main objective has been to formulate a concise query that effectively represents the underlying information need.

The rest of the paper is organized as follows. Section 2 describes the approach we have taken, specifically how the query is built and what experiments we designed; the runs we submitted are explained in Sect. 3 and the results are analysed in Sect. 4. Finally, in Sect. 5 we expose our conclusions and discuss future work.

[1] Further information is available in [6].

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 418–425, 2010.

2 Approach Taken

The track requires that retrieval is performed at patent level but provides several documents per patent. We decided to work with an index built at document level and then post-process the result in order to obtain a ranking of patents (each patent receives the score of its highest ranked document). This follows the intuition that the patent document that is the most similar to the query patent reflects well the connection between the query and the underlying patent.

The index we used[2] was built from all the textual fields of a query patent, i.e. invention-title, abstract, description and claims. Although the documents contain terms from three different languages, no language-oriented distinction was made at index construction time. This means that the index contains all terms in any language for each patent document. Furthermore, stemming was not applied and an English stopword list (with 733 stopwords) was used in order to remove common words. This makes sense because almost 70% of the data was written in English.

2.1 Query Formulation

A query patent contains about 7500 terms on average and, therefore, using them all would yield to high query response times. Furthermore, there are many noisy terms in the document that might harm performance. A good processing of the query patent document is a key factor in order to achieve good effectiveness.

Our experiments focused on extracting the most significant terms from the query patent, i.e. those terms that are discriminative. To this aim, we used *inverse document frequency* (idf). In our training, we concentrated on deciding the number of terms that should be included into the query. We ran this process in both a language-independent and language-dependent way (i.e. a single ranking of terms vs. three rankings of terms, one for each language).

The number of query terms is difficult to set because few query terms make that the query processing is fast but the information need might be misrepresented; on the other hand, if many terms are taken the query will contain many noisy terms and the query processing time might be prohibitive. We have studied two methods to choose a suitable number of terms: (i) establishing a fixed number of terms for all queries and (ii) establishing a fixed percentage of the query patent length. Observe that those terms that appear several times in a query patent have been considered only once in the final selection. Because of this, both the number of terms to extract and the query patent length refer to the number of unique terms.

Once the number of query terms has been selected, we must determine how they are extracted. We explored two strategies: language-independent and language-dependent. Suppose that we select n terms from the original query patent regardless of the language. This means that all query patent terms (English, French and

[2] We deeply thank the support of Erik Graf and Leif Azzopardi, from University of Glasgow, who granted us access to their indexes [2].

German terms) are ranked together and we simply select the n terms with the highest *idf* from this list. Because of the nature of the languages, it is likely that the three languages present different *idf* patterns. Besides, there are fewer German/French documents than English documents in the collection and, therefore, this introduces a bias in terms of *idf*. We therefore felt that we needed to test other alternatives for selecting terms. We tried out an extraction of terms where each language contributes with the same number of terms. In this second strategy we first grouped the terms of a query patent depending on their language (no classification was needed since every field in the XML is tagged with language information). Next, the highest $n' = \lfloor n/3 \rfloor$ terms from each group are extracted. The query is finally obtained by compiling the terms from the three groups.

In Sect. 2.3 we will explain how different configurations combining these strategies behave in terms of performance.

2.2 Retrieval Model

Initially, we used the well-known BM25 retrieval model [5] with the usual parameters ($b = 0.75$, $k_1 = 1.2$, $k_3 = 1000$). However, as shown below, we also tested several variations for b and k_1 in the submitted runs (in order to check the stability of the model w.r.t. the parameter setting).

The platform under which we executed our experiments was the Lemur Toolkit[3]. All experiments were run in the Large Data Collider (LDC), a supercomputer offered by the Information Retrieval Facility (IRF). This system, with 80 Itanium processors and 320GB of random access memory, provides a suitable environment for large-scale experiments.

2.3 Training

With the training data provided by the track, we studied two dimensions: query length and language. Query length refers to the way in which query size is set. As argued above, this can be done in a query-dependent (i.e. a given percentage of the query patent terms are selected) or query-independent way (i.e. a fixed number of terms are selected for all queries). The language dimension reflects the way in which terms are ranked (language-independent, i.e. a single rank for all terms; language-dependent, one rank of terms for every language). Hence, our training consisted of studying how the four combinations of these dimensions perform in terms of three well-known performance measures: MAP, Bpref and P10.

The results were obtained with the large training set (500 queries) of the main task, which contains queries in the three languages. We used the usual parameters for the BM25 retrieval model.

Figures 1(a), 2(a) and 3(a) report results for the case where the number of terms is fixed for all queries. Surprisingly, we get better performance when the language is not taken into account. However, Figs. 1(b), 2(b) and 3(b),

[3] http://www.lemurproject.org/

where terms are selected using a percentage of the query length, show a different trend. Figures 1(b) and 3(b) show that no significant difference can be established in terms of MAP and P10, respectively. In contrast, Fig. 2(b) shows that the language-dependent choice is slightly more consistent than the language-independent one in terms of Bpref.

Summing up, there are two competing configurations that perform well: a) the model that consists of combining the query-dependent and language-dependent strategies, and b) the model that considers the query-independent and language-independent strategies together. If we observe carefully the plots we will note that these two models do not differ much in MAP and P10 values but, in terms of Bpref, the model that is language and query-dependent performs the best. Furthermore, according to this training, we can state that a 40% of query length is a good trade-off between performance and efficiency. We therefore fixed the query-dependent and language-dependent as our reference model.

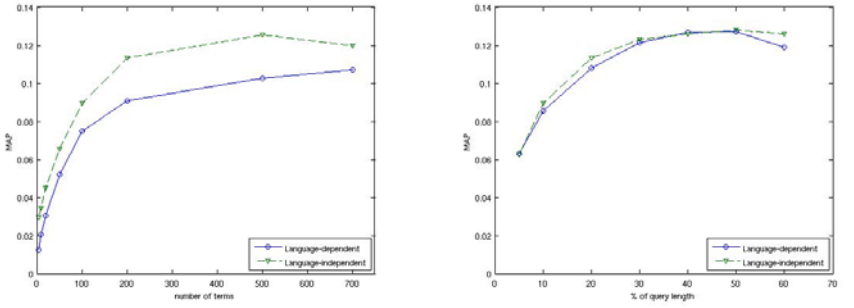
To further check that our final query production strategy is actually selecting good terms, we compared it against a baseline method. The baseline we used consists of the same retrieval system with no query formulation strategy. In this case, the queries were built by appending all the textual fields of the patents (invention-title, abstract, description and claims). This leads to very long queries with no term selection (and many terms appear more than once). Table 1 shows that our approach outperforms the baseline for each measure. So, we achieve better performance and, additionally, the query response time is expected to be significantly lower by selecting those terms that we consider more important.

Table 1. Query formulation improvement over a baseline with no term selection

	avg(#terms/query)	MAP	BPREF	P10
baseline	5656.270	.1077	.4139	.0834
query formulation	439.128	**.1269**	**.5502**	**.1012**
Δ%		+17,83%	+32,93%	+21,34%

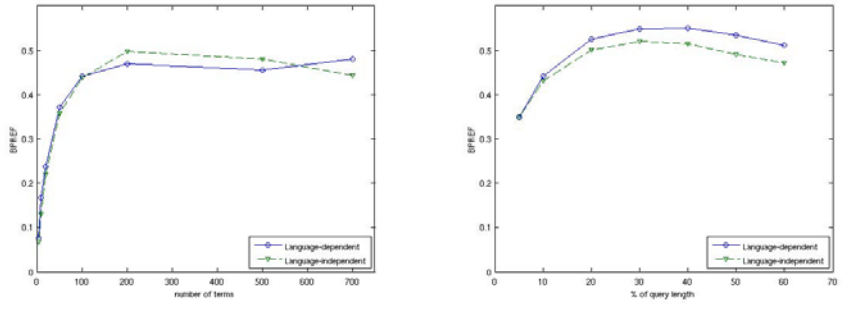
BM25 Parameters. In parallel, we performed some experiments to tune the BM25 parameters b and k_1. To this aim, we chose the small training set (5 queries) with the query-independent (500 terms) and the language-independent strategies. First, we tried several values for b keeping the other parameters fixed ($k_1 = 1.2$, $k_3 = 1000$). The observed results are described in Table 2. On the other hand, we studied the effect of the k_1 parameter for two different values of b: the recommended one ($b = 0.75$) and the value yielding the best MAP performance ($b = 1$). Again, we used $k_3 = 1000$. Tables 3 and 4 report the results.

According to this data, if we want to promote BPREF measure we should choose low values for both b and k_1. MAP, however, reaches its best performance at different places, specifically for $b = 0.75$ and $k_1 = 1.6$.



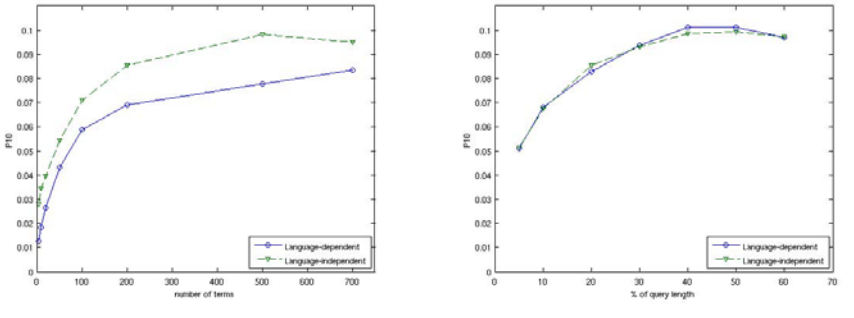
(a) Query-independent experiments (b) Query-dependent experiments

Fig. 1. MAP performance



(a) Query-independent experiments (b) Query-dependent experiments

Fig. 2. BPREF performance



(a) Query-independent experiments (b) Query-dependent experiments

Fig. 3. P10 performance

Table 2. b tuning

b	MAP	BPREF	P10
0.1	.0708	.6184	.0600
0.2	.0952	**.6302**	.0800
0.3	.1071	.6265	.0800
0.4	.1129	.6229	.1200
0.5	.1397	.6193	**.1400**
0.6	.1422	.6120	**.1400**
0.7	.1470	.5995	**.1400**
0.8	.1445	.5995	**.1400**
0.9	.1442	.5922	**.1400**
1.0	**.1529**	.6047	.1200

Table 3. k_1 tuning ($b = 0.75$)

k_1	MAP	BPREF	P10
0.2	.1020	**.6302**	.1000
0.4	.1120	.6265	.1000
0.6	.1154	.6229	.1200
0.8	.1408	.6120	**.1400**
1.0	.1400	.6120	**.1400**
1.2	.1470	.5995	**.1400**
1.4	.1465	.5959	**.1400**
1.6	**.1591**	.5922	**.1400**
1.8	.1555	.6047	**.1400**
2.0	.1513	.6047	.1200

Table 4. k_1 tuning ($b = 1$)

k_1	MAP	BPREF	P10
0.2	.1067	**.6302**	.1000
0.4	.1130	.6229	.1200
0.6	.1412	.6120	**.1600**
0.8	.1459	.5995	.1400
1.0	.1446	.5995	.1400
1.2	.1529	.6047	.1200
1.4	**.1532**	.6172	.1400
1.6	.1471	.6136	.1400
1.8	.1449	.6099	.1200
2.0	.1445	.6027	.1200

3 Submitted Runs

We participated in the *Main* task of this track with eight runs for the *Small* set of topics, which contains 500 queries in different languages.

First, we submitted four runs considering the scenario that best worked for our training experiments. These four runs differ on the retrieval model parameters. We included the recommended BM25 configuration but also tried out some variations in order to incorporate the trends that were detected in Sect. 2.3: *uscom_BM25a* ($b = 0.2$, $k_1 = 0.1$, $k_3 = 1000$), *uscom_BM25b* ($b = 0.75$, $k_1 = 1.2$, $k_3 = 1000$), *uscom_BM25c* ($b = 0.75$, $k_1 = 1.6$, $k_3 = 1000$) and *uscom_BM25d* ($b = 1$, $k_1 = 1.2$, $k_3 = 1000$).

Furthermore, we submitted four additional runs where the queries were expanded with the title terms of the query patent. In this way, the query term frequency of these terms is augmented and the presence of the title terms in the final queries is guaranteed. These new runs are labeled as the previous ones plus an extra "t".

4 Results

The official evaluation results of our submitted runs are summarized in Table 5. For each run and measure, we show both the value we got and its position among the 48 runs submitted by all groups.

The first conclusion we can extract from the evaluation is that our decision to force the presence of title terms worked well. Regardless of the configuration of the BM25 parameters, the run with the title terms always obtains better performance than its counterpart.

Furthermore, among the configurations with the title terms the best run is the one labeled as *uscom_BM25bt*. This run corresponds to the usual parameters of the BM25 retrieval model, i.e. $b = 0.75$, $k_1 = 1.2$, $k_3 = 1000$.

On the other hand, Table 6 shows another view on how good our runs performed in the evaluation. For each measure, we compare our best run with the best run and the median run in the track.

Table 5. Submitted runs for CLEF-IP 09

	P	P5	P10	P100	R	R5	R10	R100	MAP	nDCG
uscom_BM25a	.0029	.0948	.0644	.0141	.4247	.0900	.1183	.2473	.0837	.4466
	#36	#31	#32	#38	#39	#30	#32	#36	#30	#12
uscom_BM25b	.0041	.1184	.0858	.0205	.5553	.1082	.1569	.3509	.1079	.4410
	#13	#11	#6	#12	#22	#12	#6	#11	#7	#15
uscom_BM25c	.0042	.1180	.0858	.0206	.5563	.1104	.1564	.3504	.1071	.4341
	#9	#12	#7	#10	#20	#8	#7	#13	#9	#20
uscom_BM25d	.0042	.1188	.0852	.0206	.5630	.1113	.1558	.3500	.1071	.4346
	#10	#10	#9	#11	#18	#6	#8	#14	#10	#18
uscom_BM25at	.0031	.1004	.0680	.0151	.4549	.0937	.1223	.2637	.0867	.4331
	#32	#25	#31	#35	#35	#22	#30	#34	#26	#21
uscom_BM25bt	.0042	**.1280**	**.0908**	**.0213**	.5729	**.1176**	**.1631**	**.3610**	**.1133**	**.4588**
	#11	#3	#3	#4	#15	#2	#2	#5	#3	#6
uscom_BM25ct	.0042	.1268	.0898	.0212	.5722	.1172	.1611	.3602	.1132	.4544
	#12	#4	#4	#6	#16	#3	#3	#7	#4	#8
uscom_BM25dt	**.0043**	.1252	.0892	**.0213**	**.5773**	.1163	.1606	.3609	.1121	.4455
	#8	#5	#5	#5	#14	#4	#4	#6	#5	#13

Table 6. Comparative results for CLEF-IP 09

	P	P5	P10	P100	R	R5	R10	R100	MAP	nDCG
best run	.0431	.2780	.1768	.0317	.7588	.2751	.3411	.5800	.2714	.5754
our best run	.0043	.1280	.0908	.0213	.5773	.1176	.1631	.3610	.1133	.4588
	#8	#3	#3	#4	#14	#2	#2	#5	#3	#6
median run	.0034	.1006	.0734	.01785	.53535	.09275	.1309	.30855	.0887	.42485

Note that our run is ranked third for the reference measures P10 and MAP. However, in absolute terms, our results are not comparable to the results achieved by the best run. Actually, regardless of the measure, there is always a large gap between the top 1 run and the remaining ones. The first position was recurrently ocuppied by the team from Humboldt University, with their *humb_1* run [4]. Among the techniques they applied we can highlight the usage of multiple retrieval models and term index definitions, the merging of different results (and the posterior re-ranking) based on regression models and the exploitation of patent metadata[4] for creating working sets. They used prior art information from the *description* field (patents explicitly cited) as the seed of an iterative process for producing the working set. In the future, it would be interesting to compare the systems according to how good they retrieve *hidden* patents (not explicitly mentioned in the *description*).

5 Conclusions and Future Work

We have designed a query production method that outperforms a baseline with no query formulation and ranked among the top three systems for most performance measures. This method selects a number of terms that depends on the length of the original query and forces a fixed number of terms per language.

[4] The problem of comparing results based on text retrieval and re-ranking and filtering methods based on utilization of meta-data has been also outlined in [3].

The original query patent has much noise that adversely affects retrieval performance. An appropriate method for estimating the importance of the terms should be designed and applied to the patent query in order to remove noise. Nevertheless, prior art search is a recall-oriented task and reducing the query too much may harm recall.

This was our first participation in CLEF and we did not pay much attention to the cross-language retrieval problem. In the near future, we want to conduct research in this direction. We will study how to separate the patent contents by language, maintaining different indexes, etc. Furthermore, we would like to experiment with link analysis, entity extraction and structured retrieval.

Acknowledgements. We are deeply grateful to Erik Graf and Leif Azzopardi, from University of Glasgow, for their help during our experiments. We also thank the support of the IRF. This research was co-funded by FEDER and *Xunta de Galicia* under projects 07SIN005206PR and 2008/068.

References

1. Graf, E., Azzopardi, L.: A methodology for building a patent test collection for prior art search. In: Proceedings of the Second International Workshop on Evaluating Information Access, EVIA (2008)
2. Graf, E., Azzopardi, L., van Rijsbergen, K.: Automatically generating queries for prior art search. In: Working Notes for the CLEF 2009, Workshop (2009)
3. Kando, N.: Overview of the fifth NTCIR workshop. In: Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access (2005)
4. Lopez, P., Romary, L.: Multiple retrieval models and regression models for prior art search. In: Working Notes for the CLEF 2009, Workshop (2009)
5. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3, pp. 109–126 (1996)
6. Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: Retrieval experiments in the intellectual property domain. In: Multilingual Information Access Evaluation. Text Retrieval Experiments, vol. I (2010)

UAIC: Participation in CLEF-IP Track

Adrian Iftene, Ovidiu Ionescu, and George-Răzvan Oancea

UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
{adiftene,ovidiu.ionescu,george.oancea}@info.uaic.ro

Abstract. The CLEF-IP track was launched in 2009 to investigate IR techniques for patent retrieval, as part of the CLEF 2009 evaluation campaign. We built a system in order to participate in the CLEF-IP track. Our system has three main components: a filtering module, an indexing module, and a search module. Because the process of indexing all of the 75 GB of input patent documents took almost one day, we decided to work in a peer-to-peer environment with four computers. The problems encountered were related to the identification of relevant fields to be used for indexing and in the search processes.

1 Introduction

The CLEF-IP[1] (Intellectual Property) was a new track in CLEF 2009. As corpora, the IP track used a collection of more than 1M patent documents from EPO sources. The collection covered English, French and German languages with at least 100,000 documents for each language.

There were two types of tasks in the track:

- The main task was to find patent documents that constitute *prior art* to a given patent.
- *Three facultative subtasks* that used parallel monolingual queries in English, German, and French. The goal of these subtasks was to evaluate the impact of language on retrieval effectiveness.

Queries and relevance judgments were extracted using two methods: the first method used queries produced by Intellectual Property Experts and reviewed by them in a fairly conventional way, while the second was an automatic method using patent citations from seed patents. Search results reviewed ensured that the majority of test and training queries produce results in more than one language. The first results reported retrieving results across all three languages.

The way in which we built the system for the CLEF-IP track is presented in section 2, while section 3 presents the run submitted. The last section presents conclusions regarding our participation in CLEF-IP 2009.

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

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 426–429, 2010.

2 The UAIC System

Our system has three main modules: module one was responsible for the extraction of relevant fields from XML files, module two indexed the relevant fields, and the third module did the searching. Figure 1 presents the system architecture.

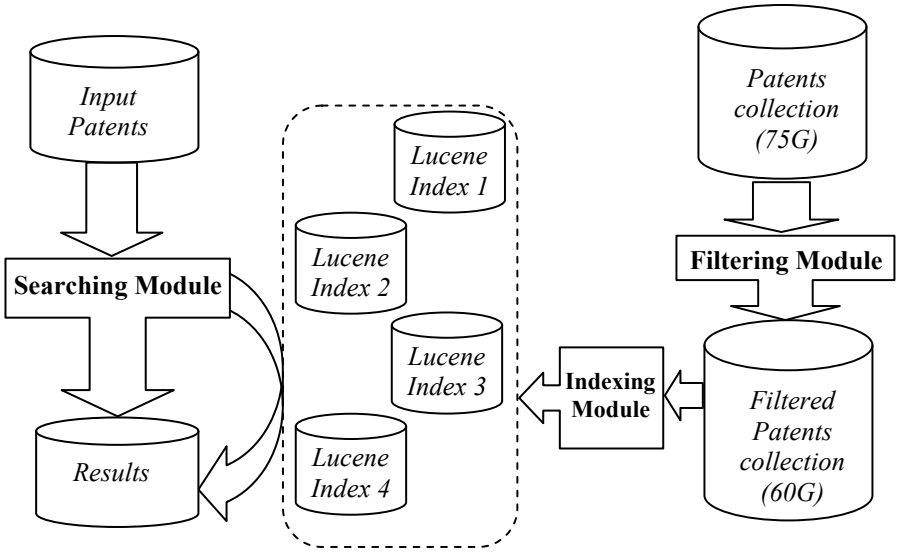


Fig. 1. UAIC system

The main components of the system (filtering, indexing and searching) worked in a distributed environment in order to reduce the processing time. Thus, for the filtering stage we used 20 computers that process, in parallel, the 75 GB of patent files and which returned the result in 3 days (instead of using only one computer for an estimated 60 days). Similarly, for indexing we used four computers that indexed 60 GB of data in approximately 6 hours (instead of using only one computer for an estimated one day). In what follows we give details about these modules.

2.1 The Extraction of Relevant Fields

The aim of this filtering step is to reduce the size of XML files that must be indexed and to only work with tags that we think are useful for this experiment. *Why*? Our initial tests demonstrate that in order to index ~150Mb we need around 94 seconds. After removing the irrelevant fields the time was reduced to 82 seconds.

The tags we agreed to keep are <*invention-title*>, <*claim text*> and <*abstract*>. If the <*abstract*> tag cannot be found we search and keep the <*description*> tag. These fields were determined after a review of some initial pre-processed documents.

2.2 The Index Creation

In order to index the corpora we used Lucene [1] a suite of free libraries used for indexing and searching large corpora. For each XML document extracted at the previous step we add to the index the fields mentioned in section 2.1. To do this we use an adapted version of the Lucene Indexer.

Because the process of indexing of all 75 GB documents took too much time (for our initial indexing we needed around 80 minutes) we decided to work in a peer-to-peer environment. Thus, we split the initial corpora on four machines and created separate indexes for each of them. These indexes were used to run queries in a parallel mode. In this way the time necessary for indexing was reduced to 20 minutes.

2.3 The Search Component

This component allows the searching of the indexes described in section 2.2. Starting from the query topics we received, we extracted the same tags used for indexing and built a Lucene query in order to search in the Lucene index. After receiving partial results from searches on the four indexes, we combined and ordered them based on their Lucene score.

When we created the Lucene query from the input patent we used different boost factors manually tuned for each specific tag. First of all, the boost values depend on the tag name. Thus, boost values are in descending order as follows: *invention-title* (2), *claim text* (1.7), *abstract* (1.4) and *description* (1) tags. Secondly, we consider two cases:

i. Higher boost values (1.2 multiplied with the above values) are used when we find one tag from topic patent in the same corresponding field from Lucene index. (e.g. for words from the *invention-title* tag from the topic patent the boost value is *1.2* x *2* = *2.4*).
ii. Lower boost values (0.8 multiplied by the highest value in the above values) are used when we have cross-searches between tags from the patent and the fields from the index. (e.g. the boost value, when the system searches for words from the *invention-title* tag from the topic patent in the *abstract* field from the index, is *0.8* x *max(2, 1.4)* = *1.6*).

The purpose of these values is to boost the score for retrieved patents which have similar values in the same document fields, but to also retrieve patents that have those values in other tags.

3 Submitted Run

Fourteen groups submitted 70 runs for this track. We submitted one run, with English as source and target language for small size proposed task. Details from the official evaluation are given in Table 1.

Participants who had better results than ours, used different filtering or indexing methods for the XML files with patents, such as using the most-up-to-date version of the xml file [4], building a full index at the lemma level [3], or creating meta-data for

each document (e.g. the first version of title, the first version of the description) [2], or using Lemur indexing (index format with patent ID and the claims section) [5].

Table 1. Official results for UAIC run

Run ID	P	R	MAP	nDCG
UAIC_MethodA	0.0004	0.0670	0.0094	0.1877

From all 500 topics we fully resolved only one (see Table 2). This was the topic for EP1291478 file, which is a patent for *a lock with universal mounting*. It seems that unlike a lot of patents, this contains a large number of claims and descriptions, so this could be the reason our search engine found all its related documents.

Table 2. Accuracy of Retrieving of Correct Answers from Gold

Accuracy	100%	50%	25%	10%	5%
Number of Topics	1	8	57	55	19

The best scores per topic were 25% (57 topics) and 10 % (55 topics), because a topic had, on average, 6.2 related documents and for the topics that we successfully identified we found, in most cases, only one document.

4 Summary

In CLEF-IP 2009 track our group submitted one run, with English as source and target language for small size proposed task. This system has three main components: a filtering module, an indexing module, and a search module. The *filtering module* has the aim to reduce the amount of XML files that must be indexed and to work only with relevant tags from the XML files. The *indexing module* used Lucene and, because the process of indexing of all 75 GB of documents was very time consuming, we used a peer-to-peer environment. The *search module* component performs searches on the index created at the previous step. When we created the Lucene query from the input patent we used different boost factors for different tags.

References

1. Hatcher, E., Gospodnetic, O.: Lucene in action. Manning Publications Co. (2005)
2. Gobeill, J., Theodoro, D., Ruch, P.: Exploring a Wide Range of Simple Pre and Post Processing Strategies for Patent Searching in CLEF IP 2009. In: CLEF 2009 Workshop Notes (2009)
3. Lopez, P., Romary, L.: Multiple Retrieval Models and Regression Models for Prior Art Search. In: CLEF 2009 Workshop Notes (2009)
4. Szarvas, G., Herbert, B., Gurevych, I.: Prior Art Search using International Patent Classification Codes and All-Claims-Queries. In: CLEF 2009 Workshop Notes (2009)
5. Verberne, S., D'hondt, E.: Prior Art Retrieval using the Claims Section as a Bag of Words. In: CLEF 2009 Workshop Notes (2009)

PATATRAS: Retrieval Model Combination and Regression Models for Prior Art Search

Patrice Lopez and Laurent Romary[1,2]

[1] Humboldt Universität zu Berlin - Institut für Deutsche Sprache und Linguistik
[2] INRIA
patrice_lopez@hotmail.com, laurent.romary@loria.fr

Abstract. This paper presents PATATRAS (PATent and Article Tracking, Retrieval and AnalysiS), a system realized at the Humboldt University for the IP track of CLEF 2009. Our approach presents three main characteristics:

1. The usage of multiple retrieval models and term index definitions for the three languages considered in the present track producing ten different sets of ranked results.
2. The merging of the different results based on multiple regression models using an additional training set created from the patent collection.
3. The exploitation of patent metadata and the citation structures for creating restricted initial working sets of patents and for producing a final re-ranking regression model.

The resulting architecture allowed us to exploit efficiently specific information of patent documents while remaining generic and easy to extend.

1 Motivations

A patent collection offers a rare opportunity of large scale experimentations in multilingual Information Retrieval. In the present work, we have tried to explore a few fundamental approaches that we consider crucial for any technical and scientific collection, namely: first, the exploitation of rich terminological information and natural language processing techniques; second the exploitation of relations among citations; and third, the exploitation of machine learning for improving retrieval and classification results.

We believe that the dissemination of patent information is currently not satisfactory. For facilitating the access and exploitation of patent publications as technical documentation, better tools for searching and discovering patent information are needed.

In the following, the *collection* refers to the data collection of approx. 1,9 millions documents corresponding to 1 million European Patent applications. This collection represents the prior art. A *patent topic* refers to the patent for which the prior art search is done. The *training set* refers to the 500 documents of *training topics* provided with judgements (the relevant patents to be retrieved).

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 430–437, 2010.

2 The Prior Art Task

As CLEF IP was launched for the first time in 2009, the prior art task was not entirely in line with a prior art task as normally performed in a patent office. The usual starting point of a patent examiner is an application document in only one language, with one or more IPC[1] classes and with a very broad set of claims. In the present task, the topic patents were entirely made of examined granted patents, so typically those for which less pertinent prior art documents were found by patent examiners. In addition, granted patents provide the following reliable information resulting normally from the search phase:

- The ECLA[2] classes.
- The final version of the claims, drafted taking into account clarity issues and the prior art identified by the examiner, together with a translation of the claims in the three official languages of the EPO by a skilled translator.
- A revised description which often acknowledges the most important document of the prior art which has been identified during the search phase.

On the other hand, some useful information for an examiner were not available such as patent families which relate patents from different patent systems.

Other more fundamental biases come from the fact that the evaluation was based on examiners search reports. The list of relevant documents cited in the search reports is by nature non-exhaustive and often motivated by procedural purposes. The goal of an examiner is not to find the best of all relevant documents, but a subset or a combination that will support his argumentation during the examination phase. In addition, non-patent literature was not considered in this task. These different aspects make the final results relatively difficult to generalize to a standard prior art search. We think, however, that overall, the organized task remains a good approximation and we consider that all the techniques presented in this work remain valid for standard prior art searches.

3 Patent Documents

A patent publication can be viewed as both a technical and a legal document. As a technical document, some key aspects have a major impact on prior art search.

Limits of the textual content. The textual content of patent documents is known to be difficult to process with traditional text processing techniques. As pointed out by [1], patents often make use of non-standard terminology, vague terms and legalistic language. The claims are usually written in a very different style than the description. A patent also contains non-linguistic material that

[1] International Patent Classification: a hierarchical classification of approx. 60.000 subdivisions used by all patent offices.

[2] European CLAssification: a fine-grained extension of the IPC corresponding to approx. 135 600 subdivisions, about 66 000 more than the IPC.

could be important: tables, mathematical and chemical formulas, technical drawings, etc. For so called drawing-oriented fields (such as mechanics), examiners focus their first attention only on drawings. Standard technical vocabularies remain, however, relevant for searching patent documents. The description section uses very often a well accepted technical terminology and a language much more similar to usual scientific and technical literature.

Citation structures. A patent collection is a very dense network of citations creating a set of interrelations interesting to exploit during a prior art search. The large majority of patents are continuations of previous work and previous patents. The citation relations make this development process visible. Similarly, fundamental patents which open new technological subfields are exceptional but tend to be cited very frequently in subsequent years.

The patents cited in the description of a patent document are potentially highly relevant documents. First, the examiner often confirms the applicant's proposed prior art by including this document in the search report. Second, in case the patent document corresponds to a granted patent, Rule 42(1)(b) of the European Patent Convention requires the applicant to acknowledge the closest prior art. As a consequence, the closest prior art document, sometimes an EP document, is frequently present in the description body of a B patent publication. We observed that, in the final XL evaluation set, the European Patents cited in the descriptions represent 8,52% of the expected prior art documents.

Importance of metadata. In addition to the application content (text and figures) and the citation information, all patent publications contain a relatively rich set of well defined metadata. Traditionally at the EPO, the basic approach to cope with the volume of the collection is, first, to exploit the European patent classification (ECLA classes) to create restricted search sets and, second, to perform broad searches on the titles and abstracts. Exploiting the ECLA classes appears, therefore, a solid basis for efficiently pruning the search space.

Multilinguality. The European Patent documents are highly multilingual. First, each patent is associated with one of the three official languages of application. It indicates that all the textual content of a patent will be available at least in this language. Second, granted patents also contains a high quality translation of the title and the claims in the three official languages made by professional human translators. Crosslingual retrieval techniques are therefore crucial for patents, not only because the target documents are in different languages, but also because a patent document often provides itself reliable multilingual information which makes possible the creation of valid queries in each language.

4 Description of the System

4.1 System Architecture

As explained in the previous section, there is clear evidence that pure text retrieval techniques are insufficient for coping with patent documents. Our

proposal is to combine useful information from the citation structure and the patent metadata, in particular patent classification information, as pre and post processing steps. In order to exploit multilinguality and different retrieval approaches, we merged the ranked results of multiple retrieval models based on machine learning techniques. As illustrated by Figure 1, our system, called PATATRAS (PATent and Article Tracking, Retrieval and AnalysiS), relies on four main steps:

1. the creation of one working set per patent topic for pruning the search space;
2. the application of multiple retrieval models (KL divergence, Okapi) using different indexes (English lemma, French lemma, German lemma, English phrases and concepts) for producing several sets of ranked results;
3. the merging of the different ranked results based on multiple SVM regression models and a linear combination of the normalized ranking scores;
4. a post-ranking based on a SVM regression model.

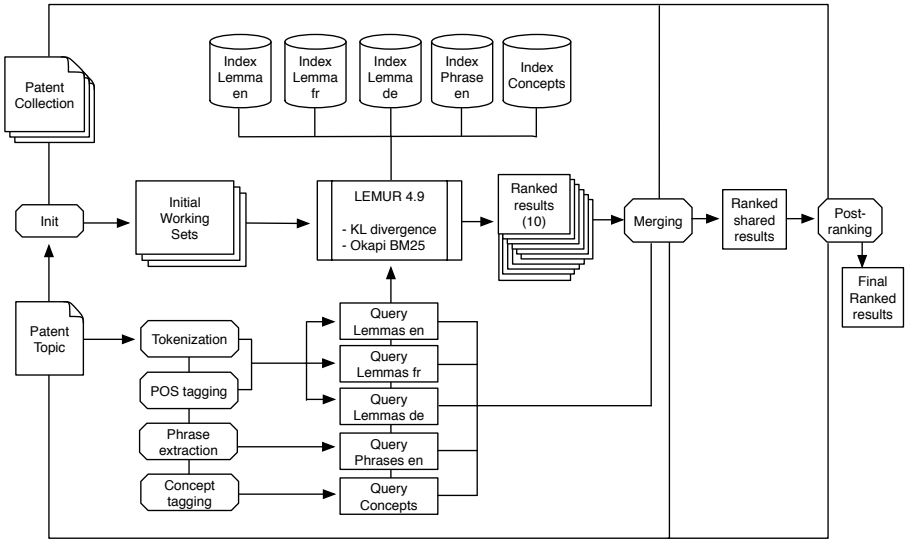


Fig. 1. System architecture overview of PATATRAS for query processing

Steps 2 and 3 have been designed as generic processing steps that could be reused for any technical and scientific content. The patent-specific information is exploited in steps 1 and 4. The next sections describe the main characteristics of our approach. For further technical details and complementary evaluations, the reader is invited to consult the full technical note [2][3].

[3] http://hal.archives-ouvertes.fr/hal-00411835

4.2 Document Parsing

All the metadata and citation relations of the patent collection have been extracted, normalized and stored in a MySQL database based on a relational model able to support intensive processing. The patents cited in the latest version of the description have been identified by means of regular expressions. For all the textual data associated with the patent, a rule-based tokenization depending on the language, a part of speech tagging and a lemmatization have been performed.

4.3 Indexing Models

We did not index the collection document by document, but rather considered a "meta-document" corresponding to all the publications related to a patent application. The five following indexes were build using the Lemur toolkit [3]:

- For each of the three languages, we built a full index at the lemma level. Only the lemmas corresponding to open grammatical categories were indexed.
- For English, we created an additional phrase index based on phrase as term definition. The Dice Coefficient was applied to select the phrases [4].
- A crosslingual concept index was finally built using the list of concepts identified in the textual material for all three languages.

4.4 Multilingual Terminological Database

A multilingual terminological database covering multiple technical and scientific domains and based on a conceptual model [5], has been created from the following existing "free" resources: MeSH, UMLS, the Gene Ontology, a subset of WordNet corresponding to the technical domains as identified in WordNet Domains [6], and a subset of the English, French and German Wikipedia restricted to technical and scientific categories. The resulting database contains approximately 3 million terms for 1,4 million concepts.

The terms of the terminological database have been used for annotating the textual data of the whole collection, training and topic sets. A term annotator able to deal with such a large volume of data has been specifically developed. The concept disambiguation was realized on the basis of the IPC classes of the processed patent. Approximately 1.1 million different terms have been identified at least once in the collection resulting in more than 176 million annotations.

4.5 Retrieval Models

We used the two following well known retrieval models: (1) a unigram language model with KL-Divergence and Jelinek-Mercer smoothing ($\lambda = 0.4$), and (2) an Okapi weighting function BM25 ($K1 = 1.5, b = 1.5, K3 = 3$). The two models have been used with each of the previous five indexes, resulting in the production of 10 lists of retrieval results for each topic patent. For both models, the query was built based on all the available textual data of a topic patent and processed

similarly to the whole collection in order to create one query per language based on lemma, one query based on English phrases and one query based on concepts. The query for a given model is, therefore, a representation of the whole textual content of the topic patent. The retrieval processes were based on the Lemur toolkit [3], version 4.9. The baseline results of the different indexes and retrieval models are presented in Table 1, column (1). On average, KL divergence performs better than Okapi, the best result being obtained with English lemma index. Concept and phrase representations suffer from information loss as compared to the simple stem-based retrievals, resulting in significantly lower performance.

Table 1. MAP results of the retrieval models, Main Task. (1) M set (1 000 queries), (2) with initial working sets, M set, (3) with initial working sets, XL set (10 000 queries).

Model: **KL divergence**

Index	Lang.	(1)	(2)	(3)
lemma	EN	**0.1068**	**0.1516**	**0,1589**
lemma	FR	0.0611	0.1159	0,1234
lemma	DE	0.0627	0.1145	0,1218
phrase	EN	0.0717	0.1268	0,1344
concept	all	0.0671	0.1414	0,1476

Model: **Okapi BM25**

Index	Lang.	(1)	(2)	(3)
lemma	EN	0.0806	0.1365	0,1454
lemma	FR	0.0301	0.1000	0,1098
lemma	DE	0.0598	0.1195	0,1261
phrase	EN	0.0328	0.1059	0,1080
concept	all	0.0510	0.1323	0.1385

4.6 Creation of Initial Working Set

For each topic patent, we created a prior working set for reducing the search space and the effect of term polysemy. The goal here is, for a given topic patent, to select the smallest set of patents which has the best chance to contain all the relevant documents. A set is created starting from the patents cited in the description of the topic patent and by applying successive expansions based on the proximity of citation relations in the global citation graph, priority dependencies, common applicants and inventors, common ECLA classes and common IPC classes. The micro recall of the final working set, i.e. coverage of all the relevant documents of the whole set of topics, was 0.7303 with an average of 2616 documents per working set. These different steps correspond to typical search strategies used by the patent examiners themselves for building sources of interesting patents. As the goal of the track is, to a large extent, to recreate the patent examiner's search reports, recreating such restricted working sets appears to be a valuable approach. Table 1, column (2) and (3) show the improvement of using the initial working sets instead of the whole collection.

4.7 Merging of Results

We observed that the different retrieval models present a strong potential of complementarity, in particular between lemmas/concepts, and would benefit from a combination. For this purpose, merging ranked results from different models and languages appears well suited for a patent collection. Moreover, we are in an exceptional situation where we can exploit a large amount of training data because

the collection contains many examples of search reports. This makes possible a fully supervised learning method. The merging of ranked results is here expressed as a regression problem [7]. Regression models appears particularly appropriate, because they permit to adapt the merging on a query-by-query basis.

For realizing the merging, the scores were first normalized. The regression model trained for a retrieval model m gives then a score for the query q which is interpreted as an estimation of the relevance of the results retrieved by m for q. The merged relevance score for a patent as prior art result for a given patent topic is obtained as a linear combination of the normalized scores provided by each retrieval model. For avoiding overfitting, we created from the collection a supplementary training set of 4.131 patents.

We have experimented several regression models: least median squared linear regression, SVM regression (SMO and ν-SVM) and multilayer perceptron using LibSVM [8] for the ν-SVM regression method and the WEKA toolkit [9] for the other methods. The best merging model was ν-SVM regression. The combined ranked result obtained with this model presents a MAP of 0.2281(+43.5% of the best individual ranked result) for the XL patent topic set, main task.

4.8 Post-ranking

While the previous section was focusing on *learning to merge ranked results*, this step aims at *learning to rank*. Regression here is used to weight a patent result in a ranked list of patents given a query. The goal of the re-ranking of the merged result is to boost the score of certain patents: patents initially cited in the description of the topic patent, patents having several ECLA and IPC classes in common, patents with higher probability of citation as observed within the same IPC class and within the set of results, patents having the same applicant and at least one identical inventor as the topic patent. Similarly as for the creation of the initial working sets, these features correspond to criteria often considered by patent examiners when defining their search strategies. The final run is also based on SVM regression, more precisely the WEKA implementation SMOreg using the standard and the supplementary training sets.

5 Final Results

5.1 Main Task

Table 2 summarizes the automatic evaluation obtained for our final runs. We processed the entire list of queries (XL set, corresponding to 10 000 patent topics), thus also covering the smaller sets (S and M, respectively 500 and 1000).

The exploitation of patent metadata and citation information clearly provides a significant improvement over a retrieval based only on textual data. By exploiting the same metadata as a patent examiner and combining them to robust text retrieval models via prior working sets and re-ranking, we created result sets closer to actual search reports. Overall, the exploitation of ECLA classes and of the patents cited in the descriptions provided the best improvements.

Table 2. Evaluation of official runs for all relevant documents (left) and for highly relevant documents (right)

Measures	S	M	XL
MAP	0.2714	0.2783	0.2802
Prec. at 5	0.2780	0.2766	0.2768
Prec. at 10	0.1768	0.1748	0.1776

Measures	S	M	XL
MAP	0.2832	0.2902	0.2836
Prec. at 5	0.1856	0.1852	0.1878
Prec. at 10	0.1156	0.1133	0.1177

6 Future Work

We have tried in the present work to create a framework that could be generalized to non-patent and mixed collections of technical and scientific documents. More complementary retrieval models and more languages can easily be added to the current architecture of PATATRAS. If some metadata are specific to patent information, many of them find their counterpart in non-patent articles.

Although our system topped the CLEF IP evaluation, we do not consider that any aspects of the present system are finalized. We plan to focus our future efforts on the exploitation of the structure of patent documents and the recognition of entities of special interest such as non patent references and numerical values.

References

1. Krier, M., Zaccà, F.: Automatic Categorisation Applications at the European Patent Office. World Patent Information 24, 187–196 (2002)
2. Lopez, P., Romary, L.: Multiple retrieval models and regression models for prior art search. In: CLEF 2009 Workshop, Technical Notes, Corfu, Greece (2009)
3. University of Massachusetts and Carnegie Mellon University: The Lemur Project (2001-2008)
4. Smadja, F., McKeown, K., Hatzivassiloglou, V.: Translating collocations for bilingual lexicons: A statistical approach. Computational Linguistics 22(1), 1–38 (1996)
5. Romary, L.: An abstract model for the representation of multilingual terminological data: Tmf - terminological markup framework. In: TAMA (Terminology in Advanced Microcomputer Applications), Antwerp, Belgium (2001)
6. Magnini, B., Cavaglià, G.: Integrating Subject Field Codes into WordNet. In: Proceedings of LREC 2000, International Conference on Language Resources and Evaluation, Athens, Greece (2000)
7. Savoy, J.: Combining multiple strategies for effective monolingual and cross-language retrieval. Information Retrieval 7(1-2), 121–148 (2004)
8. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. Technical report (2001)
9. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

NLEL-MAAT at CLEF-IP

Santiago Correa, Davide Buscaldi, and Paolo Rosso

NLE Lab, ELiRF Research Group, DSIC,
Universidad Politécnica de Valencia, Spain
{scorrea,dbuscaldi,prosso}@dsic.upv.es
http://www.dsic.upv.es/grupos/nle

Abstract. This report presents the work carried out at *NLE Lab* for the *CLEF-IP 2009* competition. We adapted the *JIRS* passage retrieval system for this task, with the objective to exploit the stylistic characteristics of the patents. Since *JIRS* was developed for the *Question Answering* task and this is the first time its model was used to compare entire documents, we had to carry out some transformations on the patent documents. The obtained results are not good and show that the modifications adopted in order to use *JIRS* represented a wrong choice, compromising the performance of the retrieval system.

1 Introduction

The *CLEF-IP 2009* arises from the growing interest by different business and academy sectors in the field of *Intellectual Property* (*IP*). The task consists in finding patent documents that constitute prior art to a given patent. *Passage Retrieval* (*PR*) systems are aimed at finding parts of text that present a high density of relevant information [3]. We based our work on the assumption that the density of the information in patent documents is high enough to be exploited by means of a *PR* system. Therefore, we adapted the *JIRS PR* system to work on *CLEF-IP 2009* data.

JIRS[1] is an open source *PR* system which was developed at the *Universidad Politécnica de Valencia* (*UPV*), primarily for *Question Answering* (*QA*) tasks. It ranks passages depending on the number, length and positions of the query n-grams that are found in the retrieved passages. In our previous participations to *Question Answering* tracks within the *CLEF Campaign*, *JIRS* proved to be superior in *PR* performance [1] to the *Lucene*[2] open source system. In the following sections, we explain the main concepts of *JIRS* system and show how we adapted JIRS in order to tackle the CLEF-IP retrieval task; in Section 5 we discuss the obtained results; and finally in Section 6 we draw some conclusions.

2 Intellectual Property Task

The main task of the CLEF-IP track consists in finding the prior art for a given patent. The corpus is composed by documents from the *European Patent*

[1] http://sourceforge.net/projects/jirs/
[2] http://lucene.apache.org

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 438–443, 2010.

Organization (*EPO*)[3] published between 1985 and 2000, a total of 1,958,955 patent documents relating to 1,022,388 patents. The provided documents are encoded in *XML* format, emphasizing these sections: title, language, summary and description, in which our approach can work properly. This supposes the omission of several fields of interest, like IPC class field, and thus a significant loss of information. A total of 500 patents are analyzed using the supplied corpus to determine their prior art; for each one of them the systems must return a list of 1,000 documents with their score ranking.

3 The Passage Retrieval Engine JIRS

The *passage retrieval* system *JIRS* is based on n-grams (an n-gram is a sequence of n adjacent words). *JIRS* has the ability to find word sequences structures in a large collection of documents quickly and efficiently through the use of different n-grams models. In order to do this, *JIRS* searches for all possible n-grams of the word sequences in the collection and it gives them a weight in relation to the amount and weights of the n-grams that appear in the query. For instance, consider the two next pasages: *"...braking system consists of disk brakes..."* and *"...anti-lock braking system developed by..."*. If you use a standard search engine, like *Yahoo* or *Lucene*, to search articles related to the phrase "anti-lock braking system", the first passage would obtain a higher weight due to the occurrences of the words containig the "brak" stem. In *JIRS* the second passage is ranked higher because of the presence of the 3-gram "anti-lock braking system". In order to calculate the n-grams weight of each passage, first of all it is necessary to identify the bigger n-gram, according to the corresponding sub n-gram presents in the query, and assign to it a weight equal to the sum of all term weights. The weight of each term is set to [1]:

$$w_k = 1 - \frac{log(n_k)}{1 + log(N)} \tag{1}$$

Where n_k is the number of passages in which the term evaluated appears and N is the total number of passages in the system and k varies between 1 and the number of words in the vocabulary of the corpus.

Once a method for assigning a weight to n-grams has been defined, the next step is to define a measure of similarity between passages. The measure of similarity between a passage (d) and a query (q) is defind as follow:

$$Sim(d, q) = \frac{\sum_{j=1}^{n} \sum_{x \in Q} h(x, D_j)}{\sum_{j=1}^{n} \sum_{x \in Q} h(x, Q_j)} \tag{2}$$

Where Q is the set of n-grams of the passage that are in the query and do not have common terms with any other n-gram. The function $h(x, D_j)$, in the equation 2, returns a weight for the j-gram x with respect to the set of j-grams (D_j) in the passage and is defined by:

[3] http://www.epo.org/

$$h(x, D_j) = \begin{cases} \sum_{k=1}^{j} w_k & \text{if } x \in D_j \\ 0 & otherwise \end{cases} \quad (3)$$

A more detailed description of the system *JIRS* can be found in [2].

4 Adaptation of JIRS to the Task

The objective was to use the *JIRS PR* system to detect plagiarism of ideas between a candidate patent and any other invention described in the prior art. We hypothesized that a high similarity value between the candidate patent and another patent in the collection corresponds to the fact that the candidate patent does not represent an original invention. A problem in carrying out this comparison is that *JIRS* was designed for the *QA* task, where the input is a query: the *JIRS* model was not developed to compare a full document to another one but only a sentence (the query , preferably short, in which terms are relevant to user needs) to documents (the passages). Therefore it was necessary to summarize the abstract of the candidate patent in order to pass it to *JIRS* as query (i.e., a sequence of words). The summarization technique is based on the *random walks* method proposed by Hassan et al. [4]. The query is composed by the title of the patent, followed by the most relevant set of words extracted from the patent abstract using this method.

Consider for instance patent EP-1445166 "Foldable baby carriage", having the following abstract:

> "A folding baby carriage (20) comprises a pair of seating surface supporting side bars (25) extending back and forth along both sides of a seating surface in order to support the seating surface from beneath. Each seating surface supporting side bar (25) has a rigid inward extending portion (25a) extending toward the inside so as to support the seating surface from beneath, at a rear portion thereof. The inward extending portion (25a) is formed by bending a rear end portion of the seating surface supporting side bar (25) toward the inside."

The random walks method extracts the relevant n-gram *seating surface* from the patent abstract. The resulting query is "Foldable baby carriage, seating surface".

Another problem was to transform the patents into documents that could be indexed by *JIRS*. In order to do so, we decided to eliminate all the irrelevant information to the purpose of passage similarity analysis, extracting from each document its title and the description in the original language in which it was submitted. Each patent has also an identification number, but often the identification number is used to indicate that the present document is a revision of a previously submitted document: in this case we examine all documents that are part of a same patent and remove them from the collection. With these transformations we obtained a database that was indexed by the search engine *JIRS*, in which each of the patents was represented by a single passage. Due to the corpus

is provided in three languages, we decided to implement three search systems, one for each language. Therefore, the input query for each system is given in the language the system was developed. To translate all queries we used the *Google Translation Tool*[4]. For each query we obtained a list of relevant patents by each of the 3 search engines. Finally, we selected the 1,000 better ranked patents. The architecture of our multilingual approach is illustrated in Figure 1.

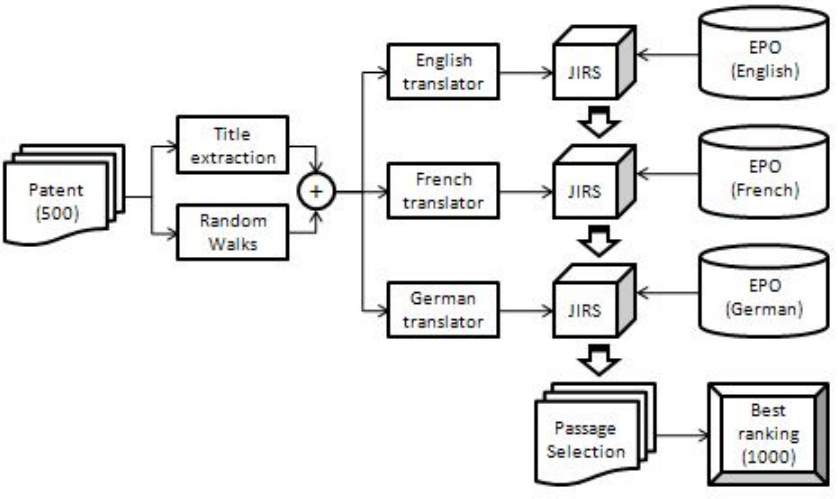


Fig. 1. Architecture of NLEL-MAAT multilingual system

5 Results

We submitted one run for the task size S (500 topics), obtaining the following results [5]:

Table 1. Results obtained for the IP competition by the *JIRS*-based system; P: Precision, R: Recall, nDCG: Normalized Discounted Cumulative Gain, MAP: Mean Average Precision

P	R	MAP	nDCG
0,0016	0,2547	0,0289	0,3377

In general, most of the results obtained by the participants were low, due to the complexity of the *IP* task. We have to emphasize that with an approach as simple as the one we have proposed, we have obtained results were not too far from the ones obtained by the best systems, the best system achieve a Precision@100

[4] http://www.google.com/language_tools?hl=en

measure of 0.0317 while our system achieve a Precision@100 measure of 0.0076. From a practical viewpoint, our aim was to apply the simple *JIRS*-based system in order to filter out non-relevant information with respect to the prior art of a patent. This allows to sensibly reduce the size of the data set to investigate eventually employing a more formal approach.

6 Conclusions

The obtained results were not satisfactory, possibly due to the reduction process carried out on the provided corpus, this allows us to be efficient in terms of performance but involves the loss of important information such as the IPC class, inventors, etc.; however we believe that the assumptions made in the approximation still constitute a valid approach, capable of returning appropriate results; in the future, we will attempt to study how to reduce the database size in order to delete as little relevant information as possible.

The development of the queries regarding each of the patents is one of the weaknesses which must be taken into account for future participations: it will be necessary to refine or improve the summarization process and to compare this model to other summarization models and other standard similarity measures between documents as well as similarity measures that include other parameters than the n-grams.

The NLP approach used in the experiments has been negatively affected by the need to use translation tools which degradated the quality of the information to be extracted from text. We believe that an approach based on the automatic categorization of documents can produce better results.

Acknowledgments. The work of the first author has been possible thanks to a scholarship funded by Maat Gknowledge in the context of the project with the Universidad Politécnica de Valencia Módulo de servicios semánticos de la plataforma G. We also thank the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 research project.

References

1. Buscaldi, D., Gómez, J.M., Rosso, P., Sanchis, E.: N-Gram vs. Keyword-Based Passage Retrieval for Question Answering. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 377–384. Springer, Heidelberg (2007)
2. Buscaldi, D., Rosso, P., Gómez, J.M., Sanchis, E.: Answering questions with an n-gram based passage retrieval engine. In: Journal of Intelligent Information Systems, (82): (Online First, 2009) ISSN: 0925-9902 (Print) 1573-7675 (Online). doi :10.1007/s10844-009-0082-y
3. Callan, J.P.: Passage-level evidence in document retrieval. In: SIGIR 1994: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 302–310. Springer, New York (1994)

4. Hassan, S., Mihalcea, R., Banea, C.: Random-Walk Term Weighting for Improved Text Classification. In: ICSC 2007: Proceedings of the International Conference on Semantic Computing, Washington, DC, USA, pp. 242–249. IEEE Computer Society, Los Alamitos (2007)
5. Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: retrieval experiments in the Intellectual Property domain. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece (2009)

Simple Pre and Post Processing Strategies for Patent Searching in CLEF Intellectual Property Track 2009

Julien Gobeill[1], Emilie Pasche[2], Douglas Teodoro[2], and Patrick Ruch[1]

[1] BiTeM group, University of Applied Sciences, 7 rte de Drize, 1227 Carouge, Switzerland
[2] BiTeM group, University and Hospitals of Geneva, Service d'Informatique Médicale, 4 rue Gabrielle-Perret-Gentil, 1211 Genève 14, Switzerland
{Julien.Gobeill,Patrick.Ruch}@hesge.ch
http://eagl.unige.ch/bitem/

Abstract. The objective of the 2009 CLEF-IP Track was to find documents that constitute prior art for a given patent. We explored a wide range of simple pre-processing and post-processing strategies, using Mean Average Precision (MAP) for evaluation purposes. Once determined the best document representation, we tuned a classical Information Retrieval engine in order to perform the retrieval step. Finally, we explored two different post-processing strategies. In our experiments, using the complete IPC codes for filtering purposes led to greater improvements than using 4-digits IPC codes. The second post-processing strategy was to exploit the citations of retrieved patents in order to boost scores of cited patents. Combining all selected strategies, we computed optimal runs that reached a MAP of 0.122 for the training set, and a MAP of 0.129 for the official 2009 CLEF-IP XL set.

1 Introduction

According to the European Patent Office (EPO), 80% of the world technical knowledge can be found in patent documents [1]. Moreover, patents are the only tool for companies to protect and take benefit from their innovations, or to check if they are free to operate in a given field or technology. As patent applicants have to provide a prior art search describing the field and the scope of their invention, and as a single missed document can invalidate their patent, patent searching is a critical field for the technical, scientific and economic worlds.

A Patent Track is proposed in NTCIR [2] since its third edition in 2002. As the NTCIR workshops took place in Japan and dealt with Asian languages, they did not retain all the attention of the Western Information Retrieval community. At the initiative of the Information Retrieval Facility, two tracks in the area of patent retrieval were launched in 2009: the CLEF-IP competition in Europe [3] and the TREC Chemistry competition in North America [4]. These tracks aim at bridging the gap between the Information Retrieval community and the world of professional patent search.

The 2009 CLEF-IP Track was defined in the official guidelines as being a prior art search task: the goal was to find patents that constitute prior art for a given patent, in a collection of patent documents from EPO sources [5]. As there were more than 1M patent documents, and as these patent documents were huge files (often several

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 444–451, 2010.

megabytes), the task was firstly to be considered as a very large scale Information Retrieval task. The preprocessing strategies hence are essential in order to work with a manageable but efficient collection. On the other hand, the different structured fields in patents make possible several post-processing strategies in different domains, such as text categorization with IPC codes, or co-citations networks with references.

Thanks to a well designed training set, with 500 patents used as queries, we were able to explore and evaluate a wide range of the strategies we mentioned above. In the following sections, we present and discuss the different strategies in the same order in which we explored them during our work on the 2009 CLEF-IP Track.

2 Data and Strategy

The CLEF-IP 2009 collection contained around 1'950'000 patent documents from the EPO. As several patent documents could belong to a same patent, there were actually around 1 million patents. Each patent document was a XML file containing structured data; different fields were delimited by specific tags. Fields that retained our attention were:

- *Title*
- *Description*: the complete description of the invention, which is the longest field.
- *Abstract*: a summary of the description field.
- *Claims*: the scope of protection provided by the patent.
- *IPC codes* : codes belonging to the International Patent Classification and describing technological areas
- *Citations*: patents cited in the prior art.

Inventor and *Applicant* fields were not retained, as we assumed they were not informative. We now think that we should have included these fields in the experiments. Moreover, we used IPC codes in two different formats: 4-digits codes (e.g. D21H) and complete codes (e.g. D21H 27/00). Citations were not used for building the patent representation, but were investigated for post processing purposes.

The task was to find patents that constitute the prior art for a given patent; in other words, participants had, from a given patent for which organizers had discarded the *Citations*, to re-build the *Citations* field. A training set of 500 patents was provided. In the *Citations* field, another patent can be cited because it can potentially invalidate the invention, or more generally because it is useful for the understanding of invention. Thus, two ways were possible in order to define what citations have to be rebuilt: a stringent qrel or a liberal qrel. All results reported in this paper were evaluated with the liberal qrel. More information is available in the official guidelines [5].

During our experiments, we explored and evaluated a wide range of strategies. Indeed, as queries can be generated only by discarding the *Citations* field, organizers were able to generate a large training set. We chose to firstly develop a complete pipeline with default settings, in order to be able to evaluate a baseline run; thus, we were able to evaluate any strategy we explored by comparing it to the baseline run. Runs were evaluated with Mean Average Precision (MAP). The Information Retrieval step was performed with the Terrier engine [6]. Thus, our approach can be seen as a gradient descent approach.

The first run we computed, with all mentioned patent fields representing the document and the queries, with standard Terrier settings and without any post-processing strategy, reached a MAP of 0.074.

3 Document and Query Representation

The first step was to decide how to merge several patent documents belonging to the same patent into a unique file. The official guidelines proposed several strategies, but we decided to keep all information contained in the different files and to concatenate it in a unique patent file.

3.1 Document Representation

The second step was to determine which fields to keep in the indexed patent files. Our priority was to keep the *Description*, as we hypothesized it would be the more informative field. However, the *Description* fields in patents are often huge, so we had to take care not to generate an unmanageable collection. Hence, our strategy was to lighten the *Description* field, by discarding the most frequent words in the collection. Experiments showed that the best performances were obtained by discarding a list of 500 stopwords, using Porter stemming. This still left us with a huge amount of data. Worst, we observed that discarding the whole *Description* field for document representation led to a MAP of 0.097, which was a + 30% improvement. Despite all our efforts, the *Description* field as we used it contained more noise than information, and we had to discard it for the patent representation. Nevertheless, we chose to keep Porter stemming and this list of 500 stopwords for pre-processing the other fields.

Table 1 shows some supplementary results on how much each field contributed to the final performance. We established a new baseline run by using all the fields except for the *Description*; the MAP for this baseline run was 0.097. Then, we discarded each field separately and observed how the MAP was affected.

Table 1. Mean Average precision (MAP) for different Document Representation strategies

Discarded field	MAP
Baseline	0.097
Title	0.096
Abstract	0.091
Claims	0.052
IPC 4-digits codes	0.0791
IPC complete codes	0.0842

Results show that the *Claims* are the most informative field, as using them led to a + 86 % improvement. This result contradicts the remarks of the patent expert provided by the official guidelines [5], that suggested that "*claims don't really matter in a prior art searches […] whereas it would be significant for validity or infringement searches*", unless the task finally must be seen as a validity search task. Another result is that the *Title* seems to be poorly informative. This result is coherent with what

Tseng and Wu wrote in their study describing search tactics patent engineers apply [7]: "*It is noted that most patent engineers express that title is not a reliable source in screening the search results […] [as] the person writing up the patent description often chooses a rather crude or even unrelated title*". Finally, we chose to keep all fields except *Description* in order to build the document representation.

3.2 Query Representation

For Query Representation, we investigated the same strategies than for Document Representation. We chose to use Porter stemming and the designed list of 500 stopwords. Then we evaluated different subsets of fields. While for Document Representation, discarding the *Description* field appeared to be the best choice, for Query Representation we obtained slightly better performances when including it (+ 3%). Hence, we chose to keep all fields in order to build the query representation.

4 Retrieval Model

Once we determined the Document and Query Representation, we tuned the Information Retrieval system in order to find the best settings. We used the Terrier 2.2.1 platform for retrieval.

Firstly, we evaluated several available weighting models in Terrier with their default settings and reached the conclusion that we didn't need to change the default BM25. We then tuned the BM25 weighting model by setting the b parameter; we finally reached a MAP of 0.105 with b=1.15. Finally, we observed that using query expansion with the available Bo1 model (Bose-Einstein inspired), set with default parameters, led to a final MAP of 0.106.

5 Post Processing Strategies

Once we determined the best retrieval model, we focused on how additional information contained in patent document could be used for re-ranking and improving the computed run. We chose to explore two different strategies: whether to filter out-of-domain patents regarding to IPC codes, or to boost related patents according to the citations of the retrieved patents.

5.1 IPC Filtering

In an expert patent searching context, Stemitzke [9] assumed in his abstract that "*patent searches in the same 4-digits IPC class as the original invention reveal the majority of all relevant prior art in patent*". Another study assumed that it is between 65% and 72% – whether citations were added by the applicant of the examiner – of European patent citations that are in the same technology class [10]. Moreover, dealing with what IPC granularity – whether 4-digits or complete codes – using in patent searches, the EPO best practices guidelines indicate that "*for national searches […] the core level is usually sufficient*" [11].

Hence, we decided to explore IPC filtering strategies that consisted in filtering (i.e. simply discarding in the ranked list) retrieved patents that did not share any IPC code with the query. We evaluated this strategy for both 4-digits and complete codes. Moreover, another strategy could consist in, for each query, only indexing documents that share at least one IPC code with the query. Thus we evaluated both strategies, respectively named *IPC filtering* and *IPC indexing* strategies, with both IPC granularities, 4-digits and complete. Results are presented in Table 2. *IPC filtering* strategy was applied in the previous baseline run that reached a MAP of 0.106.

Table 2. Mean Average precision (MAP) for different filtering strategies using IPC codes

MAP	***IPC filtering* strategy**	***IPC indexing* strategy**
Baseline	0.106	0.106
4-digits IPC codes	0.111 (+5%)	0.112 (+6%)
complete **IPC codes**	0.118 (+11%)	0.115 (+8%)

Results show that both strategies led to improvements, but none was significantly better than the other. However, the *indexing* strategy needs to re-index a specific part of the collection for each query, which is a time-consuming process. Thus we preferred to apply the *filtering* strategy. Moreover, using the complete IPC codes let to a bigger improvement than using 4-digits codes (+11% comparing to +5%). Working on the patent representation, we also observed that complete codes seemed to be more informative (see Table 1). These results, and the designed strategy for automatic prior art searches, seem to run counter to the state of the art for expert prior art searches. Finally, the availability of effective IPC automatic coders [15] to be customized for specific domains could provide promising additional information in the future.

5.2 Co-citation Boosting

Finally, we explored post-processing strategies dealing with patent citations. Few studies addressed the co-citation issue in the patent domain. Li and al. [12] used citations information in order to design a citation graph kernel; evaluating their work with a retrieval task, they obtained better results exploiting citation network rather than only direct citations.

We computed the citation network for the collection, and we explored a range of post-processing strategies, from citation graphs to weighting schemes based on the number of citations. The best strategies reached the MAP from 0.118 to 0.122 (+3%). Another interesting result was the improvement of Recall at 1000 from 0.53 to 0.63. Unfortunately, the strategies that improved the MAP used only direct citations. We never were able to design a strategy that efficiently exploited the citation network in this track.

6 Discussion

For the first year of the track, the various strategies we investigated were relatively competitive as our best runs were ranked #2. Remarkably, those results were obtained without using the citation information provided with the query. Indeed, some competitors decided to indirectly take advantage of query citations to overweight features appearing in patent sentences where citations were located. The result of such added information may have been sufficient to improve the effectiveness of the strategy, however such a task design would not faithfully model a prior art search task since it assumes that a set of prior art citations is available. The availability of such data is only meaningful for very specific user models such as patent officers, who are assessing the validity of a submitted patent. But, for most patent users, in particular patent authors, assuming the availability of such bibliographical information is unrealistic.

Now comparing with the TREC Chemistry patent track, where our official runs scored significantly higher than other submitted runs [14], as shown in Figure 1, we can observe that mean average precision (~18%) seems about +38% higher for chemistry than for domain-independent prior art search. The reported relative effectiveness of chemistry vs. unrestricted patent retrieval can at least partially be explained by the availability of large terminological chemo-informatics resources, such as PubChem or MeSHCat, which can significantly improve retrieval effectiveness by making possible to automatically normalize chemical entities, as explored in [13] thanks to the ChemTagger (http://eagl.unige.ch/ChemTagger/).

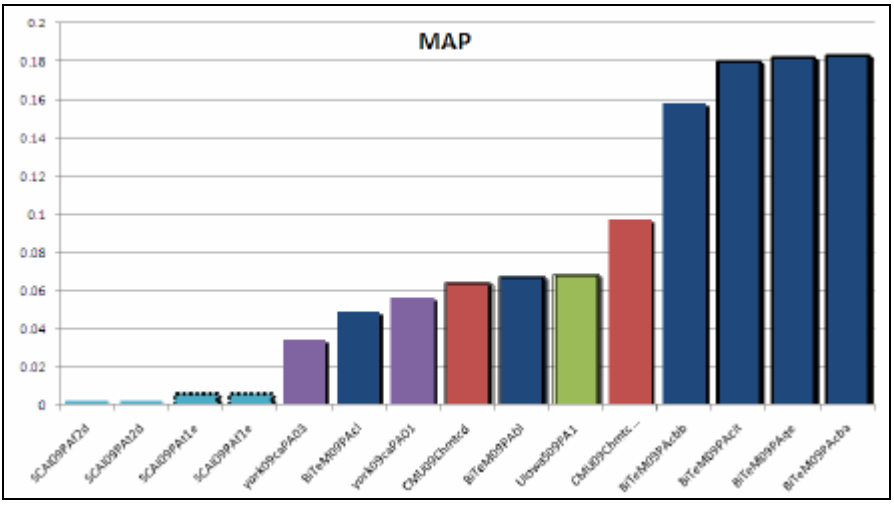


Fig. 1. Retrieval effectiveness, measured by mean average precision (map), of TREC patent competitors for chemistry. Our runs appear in blue with a map slightly above 18%.

7 Conclusion

We explored a wide range of simple strategies, aiming at choosing the best document representation, at choosing the best information retrieval platform, and at applying

some efficient post-processing tactics. Evaluated on the XL set (10'000 queries), our official run reached a MAP of 0.129. The results were satisfying, as our run was one of the leading ones. Unfortunately, strategies that improved the performances were quite simple, and we need to design more advanced winning strategies in order to still be competitive in the CLEF-IP 2010 evaluation. We probably need to improve our semantic representation of the patents, and to deal with the problem and the solution aspects of the invention. In particular, we have to pay attention to the results produced on this domain by Asian teams for the previous NTCIR competitions.

Limitations in the CLEF-IP 2009 evaluation were that retrieved documents were considered as relevant only if they were cited by the patent given as query. Yet, this does not imply that other retrieved documents were not relevant with respect the prior art of the invention. Indeed, if several documents are equally relevant to a given part of the prior art, the examiner needs to cite only one of them, choosing less or more arbitrarily. Other variables such as geographical distance, technological distance or strategic behavior of the applicant have an influence on the citations and can induce additional biases in cited patents [10]. Thus, some retrieved documents can be judged non relevant in this evaluation, because another document was chosen in the citations; but these documents could be judged relevant and useful by a professional searcher in a semi automatic process. Nevertheless, the CLEF-IP 2009 evaluation let us to start working on patent searching and to compare our strategies in a very pleasant framework.

References

1. Augstein, J.: Down with the Patent Lobby or how the European Patent Office has mutated to controlling engine of the European Economy, Diploma Thesis, University of Linz (2008)
2. http://research.nii.ac.jp/ntcir
3. http://www.clef-campaign.org
4. http://www.ir-facility.org/the_irf/trec_chem.htm
5. Piroi, F., Roda, G., Zenz, V.: CLEF-IP 2009, Track Guidelines (2009)
6. Ounis, I., Lioma, C., Macdonald, C., Plachouras, V.: Research Directions in Terrier: a Search Engine for Advanced Retrieval on the Web. Novatica/UPGRADE Special Issue on Next Generation Web Search 8, 49–56 (2007)
7. Tseng, Y.-H., Wu, Y.J.: A Study of Search Tactics for Patentability Search – a Case Study on Patent Engineers. In: Proceedings of the 1st ACM Workshop on Patent Information Retrieval (2008)
8. http://ir.dcs.gla.ac.uk/terrier/doc/configure_retrieval.html
9. Sternitzke, C.: Reducing uncertainty in the patent application procedure – insights from malicious prior art in European patent applications. World patent Information 31, 48–53 (2009)
10. Criscuolo, P., Verspagen, B.: Does it matter where patent citations come from? Inventor versus examiner citations in European patents. Research Policy 37, 1892–1908 (2008)
11. http://www.epo.org/patents/patent-information/ipc-reform/faq/levels.html
12. Li, X., Chen, H., Zhang, Z., Li, J.: Automatic patent classification using citation network information: an experimental study in nanotechnology. In: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 419–427 (2007)

13. Gobeill, J., Teodoro, D., Pasche, E., Ruch, P.: Report on the TREC 2009 Experiments: Chemical IR Track. In: TREC 2009 (2009)
14. Lupu, M., Piroi, F., Tait, J., Huang, J., Zhu, J.: Overview of the TREC 2009 Chemical IR Track. In: TREC 2009 (2009)
15. Teodoro, D., Gobeill, J., Pasche, E., Ruch, P.: Report on the NTCIR 2010 Experiments: automatic IPC encoding and novelty detection for effective patent mining. In: NTCIR 2010 (2010)

Prior Art Search Using International Patent Classification Codes and All-Claims-Queries

Benjamin Herbert, György Szarvas*, and Iryna Gurevych

Ubiquitous Knowledge Processing (UKP) Lab
Computer Science Department
Technische Universität Darmstadt
Hochschulstr. 10, D-64289 Darmstadt, Germany
http://www.ukp.tu-darmstadt.de

Abstract. In this paper, we describe the system we developed for the Intellectual Property track of the 2009 Cross-Language Evaluation Forum. The track addressed prior art search for patent applications. We used the Lucene library to conduct experiments with the traditional TF-IDF-based ranking approach, indexing both the textual content and the IPC codes assigned to each document. We formulated our queries by using the *title* and *claims* of a patent application in order to measure the (weighted) lexical overlap between topics and prior art candidates. We also formulated a language-independent query using the IPC codes of a document to improve the coverage and to obtain a more accurate ranking of candidates. Using a simple model, our system remained efficient and had a reasonably good performance score: it achieved the 6th best Mean Average Precision score out of 14 participating systems on 500 topics, and the 4th best score out of 9 participants on 10,000 topics.

1 Introduction

The CLEF-IP 2009 track was organized by Matrixware and the Information Retrieval Facility. The goal of the track was to investigate the application of IR methods to patent retrieval. The task was to perform prior art search, which is a special type of search with the goal of verifying the originality of a patent. If a prior patent or document is found that already covers a very similar invention and no sufficient originality can be proven for a patent, it is no longer valid. In the case of a patent application, this would prevent it from being granted. If a patent has already been accepted, an opposition procedure can invalidate a patent by providing references to prior art. Therefore, finding even a single prior art document can be crucial in the process, as it may have an adverse effect on the decision about patentability, or withdrawal of an application.

Prior art search is usually performed at patent offices by experts, examining millions of documents. The process often takes several days and requires

* On leave from the Research Group on Artificial Intelligence of the Hungarian Academy of Sciences.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 452–459, 2010.

strict documentation and experienced professionals. It would be beneficial if IR methods could ease this task or improve the speed and accuracy of the search.

Major challenges associated with finding prior art include: the usage of vocabulary and grammar is not enforced and depends on the authors; in order to cover a wide field of applications, often generic formulations and vague language are used; the authors might even try to disguise the information contained in a patent and take action against people that infringe a patent later; the description of inventions frequently uses new vocabulary; information constituting prior art might be described in a different language than the patent under investigation.

Dataset & Task. For the challenge, a collection of 1.9 million patent documents taken from the European Patent Office (EPO) was used. The documents in this collection correspond to approximately 1 million individual patents filed between 1985 and 2000 (thus one patent can have several files, with different versions/types of information). The patents are in the English, German, or French language. The language distribution is not uniform as 70% of the patents are English, 23% are German, and 7% are French. The patents are given in an XML format and supply detailed information such as the title, description, abstract, claims, inventors and classification.

The focus of the challenge was to find prior art for the given topic documents. Several tasks were defined: the *Main* task, where topics corresponded to full patent documents, and the multilingual tasks, where only the *title* and *claims* fields were given in a single language (*English*, *German*, or *French*) and prior art documents were expected to be retrieved in any of these three languages.

Relevance assessments were compiled automatically using the citations pointing to prior art documents found in the EPO files of the topic patent applications. The training data for the challenge consisted of 500 topics and relevant prior art. The evaluation was carried out on document sets having 500 (Small), 1,000 (Medium) and 10,000 (XLarge evaluation) topics, respectively.

For a more detailed description of the task, participating groups, the dataset and overall results, please see the challenge paper [2] and the track Web page[1].

2 Our Approach

For most patents, several files were available, corresponding to different versions of the patent (an application text is subject to change during the evaluation process). We decided not to use all the different versions available for the patent, but only the most up-to-date version. We epxected the latest version to contain the most authoritative information. If a certain field used by our system was missing from that version, we extracted the relevant information from the latest source that included this field. In our system, we used the information provided under the *claims*, *abstract*, *description*, *title* and *IPC codes* fields only. We did not use other, potentially useful sections of patent applications such as authors or date.

[1] http://www.ir-facility.org/the_irf/clef-ip09-track

2.1 Preprocessing

We performed the following preprocessing steps:

- *Sentence splitting* based on the Java BreakIterator[2] implementation.
- *Tokenization* based on the Java BreakIterator (for the French documents we also used apostrophes as token boundaries: e.g. *d'un* was split to *d* and *un*). We converted all the tokens to lowercase.
- *Stopword removal* using manually crafted stopword lists. We started with general purpose stopword lists containing determiners, pronouns, etc. for each language, and appended them with highly frequent terms. We considered each frequent word (appearing in several hundred thousand documents) a potential stopword and included it in the list if we judged it a generic term or a domain specific stopword; that is, not representative of the patent content. For example, many patents contain words like *figure* (used in figure captions and also to refer to the pictures in the text), or *invention* (it usually occurred in the 1st sentence of the documents).
- for the German language, we applied dictionary-based *compound splitting*[3].
- *Stemming* using the Porter algorithm[4].

The preprocessing pipeline was set up using the *Unstructured Information Management Architecture (UIMA)*, a framework for the development of component based *Natural Language Processing (NLP)* applications. We employed the DKPro Information Retrieval framework [1], which provides efficient and configurable UIMA components for common NLP and Information Retrieval tasks.

2.2 Retrieval

The basis of our system is the extended boolean vector space model as implemented by Lucene. We queried the indices described below and combined the results in a post-processing step in order to incorporate information gathered from both the text and the IPC codes.

Indices. In order to employ Lucene for patent retrieval, we created a separate index for each language just using fields for the relevant language. For example, to create the German index, only fields with a language attribute set to *DE* were used.

For each patent, we extracted the text of a selection of fields (*title* only, *title & claims*, *claims & abstract & description* - limited to a fixed number of words). The concatenated fields were preprocessed in the way described above. For each patent, a single document was added to the Lucene index, and the patentNumber field was added to identify the patent.

Topic documents were preprocessed in the same manner as the document collection. All the text in the *title* and *claims* fields was used to formulate the

[2] http://java.sun.com/j2se/1.5.0/docs/api/java/text/BreakIterator.html
[3] http://www.drni.de/niels/s9y/pages/bananasplit.html
[4] http://snowball.tartarus.org

queries, without any further filtering. This way our system ranked documents according to their lexical overlap with the topic patent. A separate query was constructed for each of the languages.

To exploit the IPC codes assigned to the patents, a separate index was created containing only the IPC categories of the documents. We performed retrieval based on this IPC index, and since single IPC codes usually indentify particular scientific and technological domains, this provided a language independent ranking based on the domain overlap between the query and documents.

Queries. For the main task, sample topic documents were selected that had their *title* and *claims* fields available in all three languages. Moreover, since these documents were full patent applications they contained other fields, possibly in one or more languages, but we did not use any of these additional fields.

We created a separate query for each language and ran it against the document collection index of the corresponding language. Each query contained the whole content of the two above-mentioned fields, with each query term separated by the *OR* query operator.

For the language specific tasks, only the *title* and *claims* fields of the corresponding language were made available. We performed the same retrieval step as we did for the main task, but restricted the search to the respective language index. For instance in the French subtask, we just used the French *title* and *claims* fields to formulate our query and performed retrieval only on the French document index.

To measure the weighted overlap of the IPC codes, a separate query was formulated that included all IPC codes assigned to the topic document (again, each query term *OR*-ed together).

Result Fusion. Language specific result lists were filtered in such a way that documents which did not share an IPC code with the topic were filtered out. The language specific result lists were normalized in order to make the scores comparable to each other. The result list from the IPC code index was normalized in the same way. To prepare our system output for the language specific subtasks, we added the relevance scores returned by the IPC and the textual query and ranked the results according to the resulting relevance score. For the *Main* task submission, the three language-specific lists were combined into a single list by taking the highest score from each language specific result list for each document. For further details about the result list combination, see [3].

3 Experiments and Results

In this section we present the performance statistics of the system submitted to the CLEF-IP challenge and report on some additional experiments performed after the submission deadline. We apply Mean Average Precision (MAP) as the main evaluation metric, in accordance with the official CLEF-IP evaluation. Since precision at top rank positions is extremely important for systems that are

supposed to assist manual work like a prior art search, for comparison we always give the precision scores at 1 and 10 retrieved documents (P@1 and P@10)[5].

3.1 Challenge Submission

We utilized the processing pipeline outlined above to extract text from different fields of patent applications. We experimented with indexing single fields, and some combinations thereof. In particular, we used only titles, only claims, only description or a combination of *title* and *claims* for indexing.

As the *claims* field is the legally important field, we decided to include the whole *claims* field in the indices for the submitted system. We employed an arbitrarily chosen threshold of 800 words for the indexed document size. That is, for patents with a short *claims* field, we added some text taken from their abstract or description respectively, to have at least 800 words in the index for each patent. When the claims field itself was longer than 800 words, we used the whole field. This way, we tried to provide a more or less uniform-length representation of each document to make the retrieval results less sensitive to document length. We did not have time during the challenge timeline to tune the text size threshold parameter of our system, so this 800 words limit was chosen arbitrarily – motivated by the average size of *claims* sections.

Table 1 shows the MAP, P@1 and P@10 values and average recall (over topics, for top 1000 hits) of the system configurations we tested during the CLEF-IP challenge development period, for the Main task, on the 500 training topics. These were: **1)** the system using the IPC-code index only; **2)** the system using a text-based index only; **3)** the system using a text-based index only, the result list filtered for matching IPC code; **4)** a combination of result lists of 1) and 2); **5)** a combination of result lists of 1) and 3).

Table 1. Performance on Main task, 500 train topics

Nr.	Method	MAP	P@1	P@10	avg. recall
(1)	IPC only	0.0685	0.1140	0.0548	0.6966
(2)	Text only	0.0719	0.1720	0.0556	0.4626
(3)	Text only - filtered	0.0997	0.1960	0.0784	0.6490
(4)	IPC and text	0.1113	0.2140	0.0856	0.6960
(5)	**IPC and text - filtered**	**0.1212**	**0.2160**	**0.0896**	**0.7319**

The bold line in Table 1 represents our submitted system. This configuration gave the best scores on the training topic set for each individual language. Table 2 shows the scores of our submission for each language specific subtask and the Main task on the 500 training and on the 10,000 evaluation topics.

[5] During system development we always treated every citation as an equally relevant document, so we only present such an evaluation here. For more details and analysis of the performance on highly relevant items (e.g. those provided by the opposition), please see the task description paper [2].

Table 2. Performance scores for different subtasks on training and test topic sets

	Train 500				Evaluation 10k			
Task	**MAP**	**P@1**	**P@10**	**avg. recall**	**MAP**	**P@1**	**P@10**	**avg. recall**
English	0.1157	0.2160	0.0876	0.7265	0.1163	0.2025	0.0876	0.7382
German	0.1067	0.2140	0.0818	0.7092	0.1086	0.1991	0.0813	0.7194
French	0.1034	0.1940	0.0798	0.7073	0.1005	0.1770	0.0774	0.7141
Main	0.1212	0.2160	0.0896	0.7319	0.1186	0.2025	0.0897	0.7372

3.2 Post Submission Experiments

After the submission, we ran several additional experiments to gain a better insight into the performance limitations of our system. We only experimented with the English subtask, for the sake of simplicity and for time constraints. First, we experimented with different weightings for accumulating evidence from the text- and IPC-based indices.

We found that slightly higher weight to text-based results would have been beneficial to performance in general. Using 0.6/0.4 weights, which was the best performing weighting on the training set, would have given a 0.1202 MAP score for English, on the 10k evaluation set – which is a 0.4% point improvement.

We also examined retrieval performance using different document length thresholds. Hence, we extracted the first 400, 800, 1600 or 3200 words of the concatenated claims, abstract and description fields to see whether more text could improve the results. Only a slight improvement could be attained by using more text for indexing documents. On the training set, the best score was achieved using 1600 words as the document size threshold. This would have given 0.1170 MAP score for English, on the 10k evaluation set – which is only a marginal improvement over the submitted configuration.

Previously we discarded all resulting documents that did not share an IPC code with the topic. This way, retrieval was actually constrained to the cluster of documents that had overlapping IPC codes. A natural idea was to evaluate whether creating a separate index for these clusters (and thus having in-cluster term weighting schemes and ranking) is beneficial to performance. We found that using such local term weights improved the performance of our system for each configuration. The best parameter settings of our system on the training topics were: 1600 words threshold; 0.6/0.4 weights for text/IPC indices; indexing the cluster of documents with a matching IPC code for each topic. This provided a MAP score of 0.1243, P@1 of 0.2223 and p@10 of 0.0937 on the 10,000 document evaluation set, for English. This is a 0.8% point improvement compared to our submitted system.

3.3 Significance Analysis

We used the *paired t-test* for significance tests. Due to the huge number of topics (we presented results for 10,000 topics) even small differences in MAP values tend to be statistically significant using very low significance thresholds.

First, the difference in performance between our submitted system and that of our post-submission experiments was statistically significant ($P < 10^{-4}$) even though the latter system only utilized English texts, not all three languages. This means that for our approach it was more important to set the corresponding weights for the combination, indexed text size and to use accurate term weights (in-cluster) than to exploit results for less frequently used languages. The performance of our submitted system (0.1186 MAP, placed 4th in the XL evaluation) is significantly different from both the one placed third (0.1237 MAP, $P < 10^{-3}$) and fifth (0.1074 MAP, $P < 10^{-4}$). The performance of our post-submission system (0.1243 MAP) is significantly different from the one placed second in the challenge (0.1287 MAP, $P < 10^{-2}$), but the difference compared to the third place system is not significant (0.1237 MAP, $P > 0.5$).

4 Discussion

In the previous section we introduced the results we obtained during the challenge timeline, together with some follow-up experiments. We think our relatively simple approach gave fair results, our submission came 6th out of 14 participating systems on the evaluation set of 500 topics[6] and 4th out of 9 systems on the larger evaluation set of 10,000 topics. Taking into account the fact that just one participating system achieved remarkably higher MAP scores and the simplicity of our system, we find these results promising.

We should mention here that during the challenge development period, we made several arbitrary choices regarding system parameter settings, and that even though we chose reasonably well performing parameter values tuning these parameters could have improved the accuracy of the system to some extent.

The limitations of our approach are obvious though. First, as our approach mainly measures lexical overlap between the topic patent and prior art candidates, prior art items that use substantially different vocabulary to describe their innovations are most probably missed by the system. Second, without any sophisticated keyword / terminology extraction from the topic claims, our queries are long and probably contain irrelevant terms that place a burden on the system's accuracy. Third, the patent documents provided by the organizers were quite comprehensive, containing detailed information on inventors, assignees, priority dates, etc. Out of these information types we only used the IPC codes and some of the textual description of patents. Last, since we made a compromise and searched among documents with a matching IPC code (and only extended the search to documents with a matching first three digits of IPC when we had an insufficient number of retrieved documents in the first step), we missed those prior art items that have a different IPC classification from the patent being investigated. We think these patents are the most challenging and important items to identify and they are rather difficult to discover for humans as well.

[6] Since the larger evaluation set included the small one, we consistently reported results on the largest set possible. For more details about performance statistics on the smaller sets, please see [2].

5 Conclusions and Future Work

In this study, we demonstrated that even a simple Information Retrieval system measuring the IPC-based and lexical overlap between a topic and prior art candidates works reasonably well: our system gives a True Positive (prior art) top ranked for little more than 20% of the topics. We think that a simple visualization approach like displaying content in a parallel view highlighting textual/IPC overlaps could be an efficient assistant tool for a manual prior art search (performed at Patent Offices).

In the future we plan to extend our system in several different ways. We already know that local and global term weightings behave differently in retrieving prior art documents. A straightforward extension would be therefore to incorporate both weightings in order to improve our results even further. Similarly, experimenting with other weighting schemes than the one implemented in Lucene is another straightforward way of extending our current system.

Acknowledgements

This work was supported by the German Ministry of Education and Research (BMBF) under grant 'Semantics- and Emotion-Based Conversation Management in Customer Support (SIGMUND)', No. 01ISO8042D, and by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under the grant No. I/82806.

References

1. Müller, C., Zesch, T., Müller, M.C., Bernhard, D., Ignatova, K., Gurevych, I., Mühlhäuser, M.: Flexible UIMA Components for Information Retrieval Research. In: Proceedings of the LREC 2008 Workshop 'Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP', Marrakech, Morocco, pp. 24–27 (May 2008)
2. Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: Retrieval Experiments in the Intellectual Property Domain. In: Working Notes of the 10th Workshop of the Cross Language Evaluation Forum (CLEF), Corfu, Greece (2009)
3. Szarvas, G., Herbert, B., Gurevych, I.: Prior Art Search using International Patent Classification Codes and All-Claims-Queries. In: Working Notes of the 10th Workshop of the Cross Language Evaluation Forum (CLEF) (August 2009)

UTA and SICS at CLEF-IP'09

Antti Järvelin[1], Anni Järvelin[1,⋆], and Preben Hansen[2]

[1] University of Tampere, Department of Information Studies and Interactive Media, FI-33014 University of Tampere, Finland
{antti,anni}.jarvelin@uta.fi
[2] Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden
preben@sics.se

Abstract. This paper reports experiments performed in the course of the CLEF'09 Intellectual Property track, where our main goal was to study automatic query generation from the patent documents. Two simple word weighting algorithms (modified RATF formula, and $tf \cdot idf$) for selecting query keys from the patent documents were tested. Also using different parts of the patent documents as sources of query keys was investigated. Our best runs placed relatively well compared to the other CLEF-IP'09 participants' runs. This suggests that tested approaches to the automatic query generation could be useful, and should be developed further. For three topics, the performance of the automatically extracted queries were compared to queries produced by three patent experts to see whether the automatic key word extraction algorithms seem to be able to extract relevant words from the topics.

1 Introduction

Patents are a valuable source of scientific and technological information. However, patent retrieval is a challenging task and identifying relevant patent documents among millions of international patents is time consuming, even for professional patent officers and especially for laymen applying for patents. Therefore there is a clear need for effective patent information retrieval systems. This need is emphasized in global society, where patents granted internationally may need to be found and systems capable of crossing language barriers are needed.

One of the questions that need to be solved for automatic patent retrieval to be viable is how to automatically generate queries from patent documents. Query formulation based on patent documents is a complex task and is usually carried out by patent examiners possessing extensive knowledge of both the domain of the invention and of the patent text genre. The complexity arises mostly from the language used in patent documents, which is a mixture of legislative language and domain specific technical sublanguages. The technical fields are various and the writing styles and sublanguages used in patents differ notably, which affects the term-extraction and weighting algorithms in information retrieval systems

⋆ *Current address:* Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 460–467, 2010.

[1]. While natural language texts always presents a vocabulary mismatch problem for information retrieval, the problem is compounded in patent retrieval by the frequent use of novel vocabulary, the use of legislative language in parts of patents and the intentional use of nonstandard or vague terminology by some inventors [2]. These problems suggest that automatic query generation from patent documents is not a trivial problem. Recognizing the best search concepts and keys based on their frequencies is potentially problematic. Furthermore, using query expansion seems necessary, as some central vocabulary is likely missing from the patent documents. Also, a system capable of automatically generating and running initial queries for users would certainly be useful.

Our aim when participatingi n the CLEF-IP exercise [3] was to test different approaches to automatic query generation from patent applications. We tested two simple word weighting algorithms for selecting query keys from the documents and experimented with using different parts of the patent documents as sources of query keys. The best way to analyze the performance of an automatic query generation algorithm would be a comparison against the query word selection of human experts. This is impossible in the scale of the CLEF-IP experiment, where the XL topic set consisted of 10,000 topics. We had nevertheless the opportunity to have three patent retrieval experts to analyze three of the topic documents used in the experiment. This allowed us to get a glimpse of how patent engineers might work and offered an opportunity to a better understanding of the results from the automatic runs.

The rest of the paper is organized as follows: the automatic query generation is described in Sect. 2 together with short descriptions of the manual topic analysis and of the runs. Section 3 then presents the technical details of our retrieval system. The results are presented in Sect. 4 and Sect. 5 concludes with a short discussion.

2 Query Generation

2.1 Automatic Query Generation

Two approaches to automatic query key extraction from the topics were tested: Selecting query keys based on the topic words' standard $tf \cdot idf$ weights and selecting query keys based on the topic words' Relative Average Term Frequencies in the target collection(s) and in the topics, i.e. based on the RATF formula scaled with the normalized term frequency of a topic word (for the original formula, see [4]). The modified RATF score for a key k is computed as follows:

$$RATF_{\text{mod}}(k) = \frac{tf_k}{cf_k/df_k} RATF(k) = \frac{tf_k}{cf_k/df_k} \left(10^3 \frac{cf_k/df_k}{(\ln(df_k + SP))^P} \right)$$
$$= 10^3 \frac{tf_k}{(\ln(df_k + SP))^P} \quad , \tag{1}$$

where tf_k is the frequency of the key k in a topic document, cf_k its frequency in the collection, and df_k the number of documents in which the key k occurs. Thus

the tf_k is normalized by dividing it by the logarithm of the document frequency of key k scaled with the collection dependent parameters SP and P. The runs in the CLEF-IP'09 training set suggested that the values of parameters SP and P had only minimal effect on the results.

It is not clear from which fields of the patent application documents the query keys should be selected. European patent applications include several fields with rather strictly defined contents, of which especially the text fields title, abstract, description and claims are important sources of query keys. The claims include the most salient content of a patent, i.e., the exact definition of what the invention covered by a patent is, and have therefore often been used as the primary source of query keys in patent retrieval studies [1]. Claims however are written in a legislative language, which may lead to spurious similarities between patents based on style rather than content [2]. Abstracts and descriptions include information related to the background and the field of the invention and can thus be helpful for determining the general field or class of the patent [5]. On the other hand, titles and abstracts on are sometimes written to be uninformative, while descriptions are written to be as wide as possible and may thus contain too vague terms to be used as query keys. Therefore several combinations of the text fields were tested, from using each field alone to using them all together. Only the combinations that performed best in the training runs were then tested in the official CLEF-IP runs, resulting in three different combinations of fields.

Query keys were extracted separately from the topics for all of the topic languages and for each topic three monolingual queries were formed. Also, a separate language independent query was formed of the IPC codes present at the topic documents. Each of these four queries was ran separately, the monolingual queries to the corresponding monolingual indices and the IPC query to a separate index containing only the codes, and the results were merged at a later phase. Based on training runs, using the IPC codes was not always useful in our system, and sometimes even damaged performance. Thus the IPC codes were not used in all of the final runs. In this paper, the results for four runs in XL topic set with different combinations of query key source fields and approaches to query key extraction are reported:

- All fields, IPC, RATF (UTASICS_all-ratf-ipcr)
- All fields, IPC, $tf \cdot idf$ (UTASICS_all-tf-idf-ipcr)
- Abstract and description, RATF. No IPC. (UTASICS_abs-des-ratf)
- Abstract, title and claims, IPC, RATF (UTASICS_abs-tit-cla-ratf-ipcr)

We also experimented with Google Translate for translating the missing target language patent fields from the main language's corresponding field, but these translation results are omitted due to space limitations, as the translation did not seem useful.

2.2 Manual Queries from Patent Engineers

Three English topics (EP1186311, EP1353525, EP1372507) were manually analyzed by three patent examiners (A–C) from a Swedish patent bureau to create

a baseline that the automatically generated queries could be compared to. All examiners were asked to analyze the same three patents, to get an idea of how much their analyses differ from each other. The goal was to create as realistic an initial patent query formulation situation as possible within the CLEF-IP test setting and to examine from which fields and what type of keywords and other features of the patent text the patent examiners selected. The examiners were asked to point out the ten best query keys for each topic. The query keys could be, e.g., keywords, terms, classification codes or document numbers and could be chosen from any of the patent fields. The patent examiners were also asked to write down synonyms to the selected words or other keywords that they might find useful when searching for prior art.

A set of manual queries was then formed from the ten best keywords that each patent examiner had chosen from the topics and another set so that also the synonyms pointed out by the examiners were included. Both sets contained nine queries (three examiners × three topics). Corresponding to the automatic runs, separate queries were formed of the IPC codes. The results from the natural language and IPC runs were merged at a later phase. Also "Ideal manual queries" containing all unique query words selected by examiners A–C altogether, were generated for each of the three topics. The manually generated queries varied slightly in length and contained sometimes slightly less and sometimes slightly more than ten words. The variation depended on the users being allowed to choose different types of keywords to the queries. Unlike the automatic queries, the manual queries could e.g. include phrases. The manually formed queries were compared to automatic runs, where the 10 top ranked keywords were automatically selected from all of the topic text fields using the modified RATF formula.

3 System Details

The Indri engine (KL-divergence model) of the Lemur toolkit (version 4.9) [6] was used for indexing and retrieval. The approach to indexing was to generate a single "virtual patent" of the several versions that may exist for each patent, in a manner similar to the topic generation by the organizers of the CLEF-IP'09 track (see [3]): only the central fields (title, abstract, description and claims) were indexed and only the most recent version of each of the fields was indexed. A separate monolingual index was created for each of the languages (one English, one French and one German index). The content in the different languages was extracted based on the language tags (present in the patent XML code) and a "virtual patent" was created for each language. The IPC codes were indexed separately into a language independent index. This way in total four indices were created that could be searched separately. The natural language index and the query words were stemmed using the Snowball stemmers for each language and the stop words were removed. The IPC codes were truncated after the fourth character.

As a consequence of indexing the content in each of the languages separately, three monolingual queries and an IPC code query had to be run for each of the

Table 1. Our runs compared to the best run by the Humboldt University, humb_1

Run ID	P@10	MAP	nDCG
UTASICS_all-ratf-ipc	0.0923	0.1209	**0.2836**
UTASICS_all-tf-idf-ipc	0.0930	0.1216	0.2808
UTASICS_abs-des-ratf	**0.0945**	**0.1237**	0.2759
UTASICS_abs-tit-cla-ratf-ipc	0.0838	0.1088	0.2655
humb_1	*0.1780*	*0.2807*	*0.4833*

topics. All natural language queries in all runs were set to include 50 words, based on training results. The IPC queries included all the IPC codes present in a topic document. The results from the four different queries were merged at query time separately for each topic using the MAD (Mean Average Distance) merging model developed by Wilkins et al in [7]. A detailed description of the merging approach is available in [8].

4 Results

The results are reported mainly in terms of the nDCG metric that weights the relevant documents more the higher they are ranked in the result list [9]. The nDCG results reported here were calculated with `trec_eval`-program (version 9.0) by the authors, because the nDCG results reported in the CLEF-IP track used a measure that is quite different from the standard nDCG measure. Therefore we also recalculated the nDCG results for humb_1 run reported in Table 1. For the sake of comparison, also MAP and P@10 are reported, in Table 1. The statistical significance between the differences of our runs were tested with One-way ANOVA over the nDCG-values.

The best run was the one generated from all fields, using the modified RATF formula (UTASICS_all-ratf-ipc), reaching the nDCG of 0.2836. The difference of the weakest run UTASICS_abs-tit-cla-ratf-ipc to the three other runs was statistically highly significant ($p < 0.001$). The other runs performed equally in statistical terms. Despite the modesty of the results, the runs placed relatively well compared to the other CLEF-IP'09 participants' runs. It should nevertheless be noted that the run (humb_1) by the Humboldt University (also given in Table 1) totally outclassed all others and that our best run, in terms of nDCG, reached only around 55 % of its performance.

Modified RATF and $tf \cdot idf$ performed very similarly, and both of them could be used for automatic query generation. Although our best runs (as measured with the nDCG) used all available text fields, the combination of the abstract and description fields seemed to be the most important source of query keys. Especially, abstracts were good sources of query keys.

Table 2 presents an example of the automatically and manually generated top10 queries and their overlaps computed as Jaccard similarities for the topic

Table 2. The Jaccard similarities between the queries generated from the top 10 words selected by the patent examiners and the automatically generated ten-word queries (using $RATF_{\text{mod}}$) for the topic EP1186311 (with title "Syringe pumps"). "Auto" refers to the automatically generated queries and "A", "B", and "C" to the examiner-provided queries.

Examiner	**Query**
Auto	plunger syring pump leadscrew obstruct motor actuat head encod speed
A	a61m syring motor speed rotat stop alarm obstruct
Overlap	0.31
B	a61m syring pump alarm obstruct rotat speed optic sensor detect
Overlap	0.27
C	a61m syring pump motor obstruct speed rotat slow detect fall
Overlap	0.36

EP1186311. The similarity of the queries was computed with the Jaccard similarity coefficient to be able to take the varying length of the manually created queries into account. On average, for the three topics, the Jaccard similarity between the automatic and manual queries was 0.29. On average 4.2 of the 10 query words in an automatically created query were shared with a manually created query. This means that the automatic query generation algorithm selected rather different words to the queries than the patent examiners. The examiners agreed slightly more often with each other: The Jaccard similarity between the manual queries was on average 0.39, or on average 5.1 of the query words in a manual query were shared with another manual query when the average query length of the manual queries was 9.1 words. Adding synonyms naturally reduced the overlap between the automatic and manual queries. Also the disagreement between the examiners grew as synonyms were added to the queries as the examiners rarely added the same synonyms to the queries: The average similarity between the manual synonym queries was 0.28.

Interestingly, the manually generated queries performed on average worse than the automatically generated ones. The average nDCG for the automatically generated queries was 0.3270, with considerable variation from 0.0 for the topic EP1353525 to 0.7152 for the topic EP1372507. The best manual query set (that of the examiner-C) reached an average nDCG of 0.2775 and was also the manual query set that had the highest average word overlap with the automatically generated queries. The average nDCGs for examiners A and B were 0.2381 and 0.2448 respectively. Expanding the queries with the synonyms selected by the examiners did not improve the results for the manual queries and neither did combining the manual queries into one "ideal" query – the "ideal query" performed equally with the manual query C.

All the queries retrieved almost exactly the same relevant documents. The manual queries A and B retrieved (respectively) two and one more relevant documents than the other two queries, but these were ranked so low that they did not have any real effect on the results. The differences between the queries dependent

thus only on how well they ranked the relevant documents. This suggests that even if the overlap of words in the queries was not very high, they still identified the same relevant features from the topics. The unique query words did not add new useful dimensions to the queries, but affected the ranking. The automatically generated query set reached the highest average performance mainly because it ranked the only relevant document found for the topic EP1186311 much higher than the rest of the queries. It also improved the ranking of relevant documents for the topic EP1372507 compared to the manual A and B queries.

5 Discussion

The results suggest that both the modified version of the RATF-formula and the $tf \cdot idf$ weighting could be viable alternatives for automatic query word extraction in patent retrieval. The two formulas look similar, but the modified RATF-formula down-weights words with high dfs more harshly than the $tf \cdot idf$. Based on the results, abstract seemed to be the best source of query keys. It is notable that the claims were not as good sources of query keys even though they were present in all three languages, which should have facilitated retrieval from tri-lingual collection.

Our indexing approach required post-query result merging to produce a single ranked result list. The advantage of this is that it is easy to run queries e.g. in only one of the three languages. However this also complicates the system and makes it less transparent from evaluation point of view, as the effects of the merging on the results are not known.

Automatic query generation is difficult due to the fact that not all of the necessary query words are present in the patent documents. Neither the system nor the patent examiners could identify all the search keys and concepts that were required to retrieve all the relevant documents. This means that some query expansion is needed. Fujita [10] showed that it is difficult to improve the performance of prior art searches with pseudo relevance feedback because there are typically only few relevant documents for each query and because these documents are rather different in terms of vocabulary. Finding good, up-to-date external sources of expansion keys for the fast developing technical fields is difficult, but creating a statistical expansion thesaurus based on the target collection and augmented by focused web crawling [11] of technical documents from relevant fields might be a feasible strategy. Another approach to identifying useful expansion keys from the target document collection might be using vector-based semantic analysis [12] to recognize semantically similar words based on their distribution in the collection.

It is tempting to think that the automatic query generation might have succeeded in recognizing most or at least many of the relevant features that the examiners identified from the topics. However, it has to be kept in mind that the test data of three topics was far too little to authorise any far-reaching conclusions and that the average performance for these topics was over the average in the XL topic set and the variation in performance between the topics

large. The limited comparison of manually created and automatically generated queries nevertheless suggested that the automatic queries can sometimes rank the relevant documents better than the manual queries. Therefore it seems that it could, from a user point of view, be useful to automatically expand user created queries with automatically generated queries to improve ranking in real retrieval systems.

Acknowledgments

The work of Anni Järvelin was funded by Tampere Graduate School in Information Science and Engineering (TISE), and Antti Järvelin was funded in part by the Finnish Cultural Foundation, and in part by the Academy of Finland Project No. 1124131. We wish to thank the Swedish patent bureau Uppdragshuset for their kind help and especially the three patent examiners for participating in the user assessments. The authors also wish to thank Mr Ville Siekkinen for his invaluable help on designing and building the computer used in the tests.

References

1. Mase, H., Matsubayashi, T., Ogawa, Y., Iwayama, M., Oshio, T.: Proposal of two-stage patent retrieval method considering the claim structure. ACM TALIP 4(2), 190–206 (2005)
2. Larkey, L.S.: A patent search and classification system. In: Proc. of the Fourth ACM conference on Digital Libraries, pp. 179–187. ACM, New York (1999)
3. Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: Retrieval experiments in the intellectual property domain (2009), http://clef.iei.pi.cnr.it/
4. Pirkola, A., Leppänen, E., Järvelin, K.: The RATF formula (Kwok's formula): Exploiting average term frequency in cross-language retrieval. Information Research 7(2) (2002)
5. Kim, J.H., Choi, K.S.: Patent document categorization based on semantic structural information. Inf. Process. Manage 43(5), 1200–1215 (2007)
6. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language-model based search engine for complex queries. In: Proc. of the International Conference on Intelligence Analysis (2005)
7. Wilkins, P., Ferguson, P., Smeaton, A.F.: Using score distributions for query-time fusion in multimediaretrieval. In: Proc. of the 8th ACM International Workshop on Multimedia Information Retrieval, pp. 51–60. ACM, New York (2006)
8. Järvelin, A., Järvelin, A., Hansen, P.: UTA and SICS at CLEF-IP. In: CLEF Working Notes 2009 (2009), http://clef.iei.pi.cnr.it/
9. Kekäläinen, J., Järvelin, K.: Using graded relevance assessments in IR evaluation. ACM TOIS 53(13), 1120–1129 (2002)
10. Fujita, S.: Technology survey and invalidity search: A comparative study of different tasks for Japanese patent document retrieval. Inf. Process. Manage. 43(5), 1154–1172 (2007)
11. Talvensaari, T., Pirkola, A., Järvelin, K., Juhola, M., Laurikkala, J.: Focused web crawling in the acquisition of comparable corpora. Information Retrieval 11(5), 427–445 (2008)
12. Sahlgren, M.: The Word-Space Model. PhD thesis, Stockholm University (2006)

Searching CLEF-IP by Strategy

W. Alink, Roberto Cornacchia, and Arjen P. de Vries

Centrum Wiskunde & Informatica,
Science Park 123, 1098 XG Amsterdam, Netherlands
{alink,cornacchia}@spinque.com,arjen@cwi.nl
http://www.cwi.nl/

Abstract. Tasks performed by intellectual property specialists are often ad hoc, and continuously require new approaches to search a collection of documents. We therefore investigate the benefits of a visual 'search strategy builder' to allow IP search experts to express their approach to searching the patent collection, without requiring IR or database expertise. These search strategies are executed on our probabilistic relational database framework. Search by strategy design can be very effective. We refined our search strategies after our initial submission to the CLEF-IP track, and with minor effort we could include techniques shown to be beneficial for other CLEF-IP participants.

1 Introduction

The main objective of this research is to demonstrate the importance of flexibility in expressing different strategies for patent-document retrieval. Instead of introducing new models, this paper presents our participation in CLEF-IP with a highly flexible visual environment to specify and execute search strategies interactively, while using a seamless combination of information retrieval and database technologies under the hood. The hypothesis is that expert users in the intellectual property search domain could use our tools to develop and customise complex search strategies as they see fit, without extensive training in information retrieval and databases. Carrying out complex search strategies becomes as simple as assembling basic operations on data flows visually. Visual query construction proofs particularly helpful when experimenting with a variety of search strategies. They can be easily combined, or split into separate strategies to simplify comparison of effectiveness.

Query execution within the proposed framework provides several advantages over competing solutions. Intellectual property specialists can specify arbitrary combinations of exact match operators with ranking operators, giving them a much desired degree of control. The search strategy itself, once expressed, provides an intuitive overview of the search steps used, so that results can be explained and verified. Also, defining strategies on a high level of abstraction makes it easy to modify and improve them over time, often without any programming involved.

Section 2 gives more details about the tooling support that we created. Section 3 details our participation to CLEF-IP, and provides an analysis of the

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 468–475, 2010.

short-comings of the submitted runs. Section 4 presents post-submission runs, where we use the flexibility of the strategy builder to apply easily several key ideas of other participants. Final considerations on the benefits of our approach as well as aspects for improvent are summarised in Section 5.

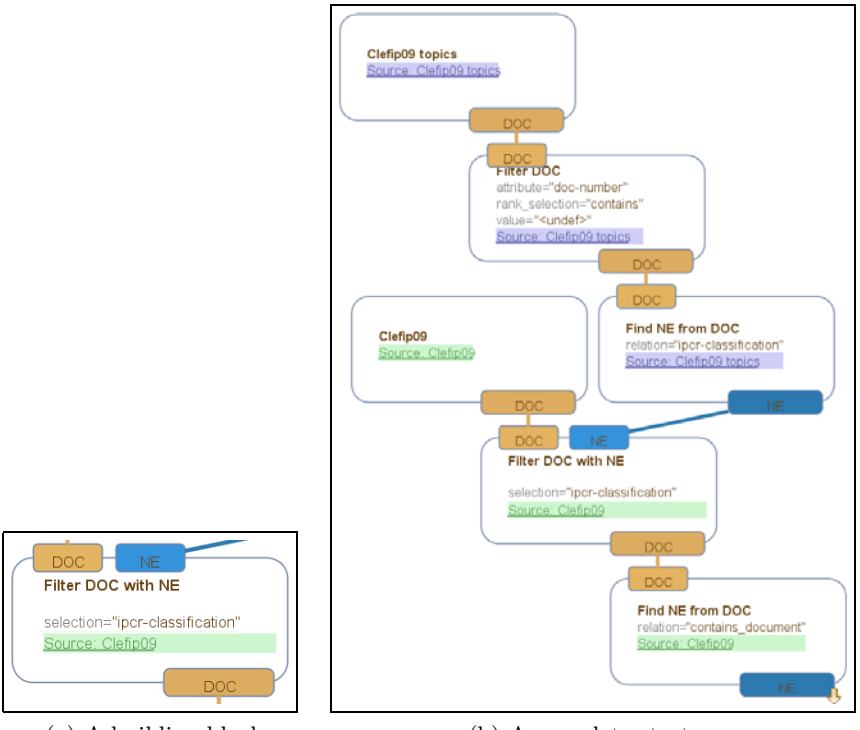
2 System Overview

Our participation in CLEF-IP was powered by the outcomes of a joint project, *LHM*, between CWI, Apriorie [7] and a leading IP search provider. The project resulted in a prototype consisting of three layers; the (1) Strategy Builder, a graphical user interface that enables intellectual property experts to create complex search strategies in a drag&drop fashion; the (2) HySpirit software framework for probabilistic reasoning on relational and object-relational data; and (3) MonetDB, an open source high-performance database management system.

2.1 Search Strategy Definition

From a data-management point of view, search strategies correspond to SQL queries over a relational database (see Section 2.3 for more information). However, the user is protected against the complexity of defining such SQL expressions. He or she assembles search strategies visually, by drag&drop of elements called *building blocks*: named boxes with input and output pins, that perform basic operations on the data in transit (such as ranking over, selection from and mixing of sources). Connections between building blocks' output and input pins represent data flows. The complete graph created by such building blocks and connections defines a visual representation of how data flows from source blocks (blocks with no input pins that identify collections) to query results (any output pin can deliver results).

Fig. 1a shows a single example building block named `Filter_DOC_with_NE`. Its first and second input pins (visualised on top of the block) accept data streams of type `DOC` (documents) and `NE` (Named Entities) respectively. This example block filters the output documents over a relation with the `NE`s in input, here selecting specific IPCR classes (performing a join operation). Fig. 1b shows how a combination of multiple building blocks defines a search strategy, in this case the strategy specifying the *category*-run submitted to CLEF-IP (returning patent-documents that match one or more IPCR classes of the topic-patent, see also Section 3). Two data sources are defined: the `Clefip2009` and the `Clefip2009 topics` collections, both delivering patent-documents. Starting from `Clefip2009 topics`, the following block performs a document selection, based on a specific document-number (defined interactively at query-time). This delivers a single document to the next block, which in turn delivers the IPCR classes (defined as of type `NE`) of that document. Such categories are the second input of the block seen in Fig. 1a. This finds all documents delivered by the `Clefip2009` collection that match the IPCR classes found at the previous step. Finally, the last block produces the strategy's result, all patents (of type `NE`) identified by the patent-documents found so far.



(a) A building block (b) A complete strategy

Fig. 1. Definition of the *category*-run strategy used for the CLEF-IP track

2.2 Search Strategy Execution

Execution of a strategy entails transformation of its visual representation into a sequence of database operations. Actions performed by building blocks are internally defined in terms of the HySpirit Probabilistic Relational Algebra (PRA) [3]. Using a probabilistic database language abstracts away the handling of probabilities, while guaranteeing the search strategy to properly propagate relevance probabilities. The first translation step collects the PRA snippets of all building blocks in a strategy and glues them together into one or more optimised PRA queries. Then, PRA specifications are translated into standard SQL queries, and executed on the high-performance database engine MonetDB [2]. Fig. 2 shows excerpts of the PRA and SQL expressions for the building block depicted in Fig. 1a. Besides being longer, the SQL query explicitly shows probability computations, nicely hidden in the corresponding PRA formulation.

2.3 Index Creation

Operations defined in building blocks rely on a relational index structure that consists of two parts: a domain-unaware index (a relational representation of the

```
BLOCK_SOURCE(docID) = INPUT1_result
BLOCK_SUBJECTS(neID) = INPUT2_result

BLOCK_prd(neID,docID)
 = PROJECT ALL [neID, docID] (
   SELECT[predicate="%SELECTION%"] (
    INPUT1_ne_doc ) );
BLOCK_nes(neID, docID)
 = PROJECT ALL [neID, docID] (
   JOIN INDEPENDENT [docID=docID] (
    BLOCK_SOURCE_result,
    BLOCK_prd ) );
BLOCK_result(docID)
 =  PROJECT DISTINCT[docID] (
   JOIN INDEPENDENT [neID=neID] (
    BLOCK_SUBJECTS_result,
    BLOCK_nes ) );
```

(a) PRA

```
CREATE VIEW BLOCK_prd_1 AS
 SELECT neID AS a1, predicate AS a2,
        docID AS a3, prob
 FROM  INPUT1_ne_doc
 WHERE predicate='%SELECTION%';

CREATE VIEW  BLOCK_prd AS
 SELECT a1, a3 AS a2, prob
 FROM  BLOCK_prd_1;

CREATE VIEW  BLOCK_nes_1 AS
 SELECT INPUT1.a1 AS a1,
   BLOCK_prd.a1 AS a2,  BLOCK_prd.a2 AS a3,
   INPUT1.prob *  BLOCK_prd.prob AS prob
 FROM  INPUT1,  BLOCK_prd
 WHERE INPUT1.a1= BLOCK_prd.a2;

CREATE VIEW  BLOCK_nes AS
 SELECT a1, a3 AS a2, prob
 FROM  BLOCK_nes_1;

CREATE VIEW BLOCK_result_1 AS
 SELECT INPUT2.a1 AS a1,
   BLOCK_nes.a1 AS a2, BLOCK_nes.a2 AS a3,
   INPUT2.prob * BLOCK_nes.prob AS prob
 FROM  INPUT2, BLOCK_nes
 WHERE INPUT2.a1=BLOCK_nes.a1;

CREATE VIEW BLOCK_result AS
 SELECT a3 AS a1, 1-prod(1-prob) AS prob
 FROM BLOCK_result_1 GROUP BY a3;
```

(b) SQL

Fig. 2. PRA and SQL expressions for the building block depicted in Fig. 1a

IR inverted file structure) and a domain-specific one (patent-specific in this case). Domain-specific knowledge is expressed as relational tuples that describe objects (e.g. patents, patent-documents, persons, IPCR classes, companies, legal events, etc), relations between objects (e.g. a person is the inventor of a patent) and object attributes (e.g. addresses, dates). Manually created XQuery expressions transform the CLEF-IP XML data into our relational schema. Index construction is carried out with MonetDB/XQuery [1] and its IR module PF/Tijah [4]. Some statistics are shown in Table 1.

3 CLEF-IP Experiments

This Section reports on the experiments conducted for the official submission, while Section 4 presents the post-submission experiments (indicated as additional runs). Fine-tuning of all parameters used, for both submitted and additional runs, was performed on the training set provided. Instead of merging patent-documents belonging to the same patent into a single document (suggested in the CLEF-IP instructions), we have indexed the original documents and aggregate scores from different patent-documents into patents as part of the search strategies.

Table 1. Index statistics

Table	Description	Size (tuples)
`tf`	term-doc frequency	850M
`termDict`	term dictionary	8.9M
`docDict`	document dictionary	6.7M
`ne_doc`	relations between named entities and documents, e.g. `(person1, inventor_of, document1)`	50M
`doc_doc`	relations between documents, e.g. `(document1, cited_by, document2)`	10M
`ne_ne`	relations between named entities, e.g. `(ipcrclass1, references, ipcrclass2)`	1.4M
`ne_string`	textual attributes for named entities, e.g. `(person1, address, 115 Bourbon Street)`	9.6M
`doc_string`	textual attributes for documents, e.g. `(document1, kind, A2)`	39M

3.1 Submitted Runs

In total 4 runs have been submitted to the CLEF-IP track, mainly dealing with standard keyword search and patent-category classifications:

***boolean*:** 10 words with the highest tf-idf scores are selected from the topic document. Patent-documents that match at least half of these are considered a match. This yields on average roughly 1000 matching patents per topic.

***bm25*:** Ranks the patent-document collection using the BM25 retrieval model [9], with a query-term list of 15 terms extracted from the topic patent-document at hand (top tf-idf scores).

***category*:** Selects patent-documents that match one or more IPCR classes of the topic-patent. IPCR classes are weighted by idf. Patent-documents are ranked by the sum of matching category scores.

***category-bm25*:** This run uses the result of the *category*-run as an input for the *bm25*-run, ranking the set of category-weighted patent-documents.

For the *boolean*, *bm25* and *category-bm25* runs, query word matching has been performed on the union of all "textual" sections of a document (title, abstract, description, and claims). The strategies produce a ranked list of patent-documents, whose scores are averaged to yield a ranked list of patents. The corpus was physically distributed over 4 different databases, each holding the index for 500k patent-documents.

3.2 Evaluation and Analysis

Results of the CLEF-IP runs have been made available in [8] and are summarised in [10].

None of our submitted runs were particularly effective when compared to other participants. Few observations can be made when looking at each run individually and when comparing them to each other: the *boolean*-run provides poor retrieval quality, as expected; the *category*-run provides high recall, which is in line with other participants' submissions; the *bm25*-run performs poorly

Table 2. Results for runs on topic set (S bundle: 500 topics)

	submitted		additional (improvement)	
Strategy	MAP	R@1000	MAP	R@1000
boolean	0.0195	0.1947		
bm25	0.0666	0.4350	0.0989 (+49.50%)	0.4984 (+14.57%)
category	0.0392	0.6656	0.0474 (+20.92%)	0.6784 (+1.92%)
category-bm25	0.0697	0.5874	0.1239 (+77.76%)	0.7408 (+26.12%)
category-bm25-cited			0.2267	0.7481

compared to similar strategies by other participants, which may indicate inaccurate parameter-tuning; the *category-bm25*-run does somewhat improve precision and recall over the *bm25*-run, but MAP remains equal; the *category-bm25*-run does somewhat improve precision and MAP over the *category*-run, but recall is lower.

We identified a number of explanations for the low effectiveness. First, the ranked list of patent-documents has been aggregated into a list of patents by averaging document scores, which turned out suboptimal. Also, we did not perform stemming, and had not merged the global statistics from the 4 collection partitions we used. Finally, in the *category-bm25* strategy we *filtered* patent-documents by IPCR class (warned against in [8]).

An interesting result, supported by other participants' results, is that patent-category information can yield a high recall, especially if patents are weighted on the number of matching IPCR classes and when the IDF of the specific IPCR class is taken into account.

4 Additional Experiments

After our official CLEF-IP submission, we investigated possible improvements over our query strategies, applying some promising ideas proposed by other participants (e.g., [6]):

***bm25*:** Number of terms extracted from topics empirically tuned to 26.
***category*:** This strategy has not been altered (see however also below).
***category-bm25*:** The combination of the two strategies is performed a little differently to avoid the empty set problem. *bm25* and *category* strategies are run independently on the original patent-document corpus and results are probabilistically merged using weights 0.9997 and 0.0003, respectively (empirically found).
***category-bm25-cited*:** This is similar to the *category-bm25* run, except that it also adds some weight to patent-documents that are cited in the topic document at hand. The weighting distribution, empirically found, is (terms: 0.9977, classification: 0.0003, citations: 0.0020).

In all additional experiments, aggregation of patent-documents into patents is performed by taking the maximum score for each group, rather than average.

Table 3. Effect of including patents from "near" IPCR classes, mixing original- (90%), children- (5%) and referenced-classes (5%)

	exact IPCR		IPCR+children+referenced	
Strategy	MAP	R@1000	MAP	R@1000
category (training-set)	0.0498	0.6361	0.0504 (+1.20%)	0.6441 (+1.26%)
category (topics)	0.0474	0.6784	0.0486 (+2.53%)	0.6791 (+0.10%)

Results with exactly equal probabilities have been sorted on descending patent-number, which resulted in more recent patents be returned first.[1] Data was physically processed and stored in a single SQL database instead of partitioned over 4 databases.

The relatively straightforward and easy to understand *category-bm25* strategy (BM25 with IPCR classification) yields results above methods by other participants, that have used far more exotic features of the corpus and topics.

In the submitted runs, the terms extracted from the topic document were weighted on the TF.IDF score (within the collection of topic documents). This caused the terms to be weighted twice on an IDF measure (once using the IDF of the topic document collection, and once in the BM25 formula using the IDF of the patent document collection). By resetting the probabilities for terms after term extraction from the topic to 1.0 (and not even taking into account the TF within the topic anymore), the double IDF weighting was prevented, and results improved. Note that for selecting the most relevant terms from the topic document TF.IDF has still been used, where IDF is estimated from the topic-document collection only.[2]

The IPCR schema organises its classes in a hierarchy, and each IPCR class may have references to related IPCR classes. Slightly better results (Table 3) can be obtained when assuming patent-documents classified with IPCR classes related to IPCR classes in the topic document (referenced- and child-classes) can be relevant. The differences are however marginal.

5 Conclusion

Regarding flexibility, visual strategy construction proved to be flexible enough to express, execute and improve several retrieval strategies on the CLEF-IP 2009 collection. In particular, combining exact and ranked matches and properly propagating probabilities required no effort. The rather simple *bm25-category* strategy shows scores comparable with more complex approaches. However, finding

[1] This modification, while only meant to be for consistency, actually improved the MAP and R@1000 a little for the *category* runs, where results often had equal probabilities.

[2] Ideally the IDF of the corpus, and some of the TF score of the extracted terms in the topic document should be taken into account, but at present this is difficult to express in the visual environment.

the right mixture values (for example between *bm25* scores and *category* scores) calls for better retrieval model self-tuning support.

The CLEF-IP track has been set up using automated systems. In a real-world setting, the intellectual property expert would guide the various stages of the process. As suggested in [5] query term refinement and expansion by experts can lead to completely different query term lists, which may result in higher quality. Note that search strategies can be executed with a user in the loop, e.g. when placing an interactive feedback building block between the term extraction from the topic patent and the BM25 search.

Acknowledgements. We thank the IRF for providing computing resources, and the MonetDB kernel developers and Apriorie for their software support.

References

1. Boncz, P., Grust, T., van Keulen, M., Manegold, S., Rittinger, J., Teubner, J.: MonetDB/XQuery: a fast XQuery processor powered by a relational engine. In: SIGMOD 2006, pp. 479–490 (2006)
2. CWI. MonetDB website, http://www.monetdb.nl/, (accessed Feburary 28, 2010)
3. Fuhr, N., Rölleke, T.: A probabilistic relational algebra for the integration of information retrieval and database systems. ACM TOIS 15(1), 32–66 (1997)
4. Hiemstra, D., Rode, H., van Os, R., Flokstra, J.: PF/Tijah: text search in an XML database system. In: OSIR (2006)
5. Järvelin, A., Järvelin, A., Hansen, P.: UTA and SICS at CLEF-IP. In: CLEF Working Notes (2009)
6. Lopez, P., Romarya, L.: Multiple retrieval models and regression models for prior art search. In: CLEF Working Notes (2009)
7. Apriorie LTD. Apriorie website, http://www.apriorie.co.uk/ (accessed Feburary 16, 2010)
8. Piroi, F., Roda, G., Zenz, V.: CLEF-IP 2009 Evaluation Summary (2009)
9. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Third Text REtrieval Conference, TREC 1994 (1994)
10. Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: retrieval experiments in the Intellectual Property domain. In: CLEF Working Notes (2009)

UniNE at CLEF-IP 2009

Claire Fautsch and Jacques Savoy

Computer Science Department, University of Neuchatel,
Rue Emile Argand 11, 2009 Neuchâtel, Switzerland
{Claire.Fautsch,Jacques.Savoy}@unine.ch

Abstract. This paper describes our participation to the Intellectual Property task during the CLEF-2009 campaign. Our main objective was to evaluate different search models and try different strategies to select and weight relevant terms from a patent to form an effective query. We found out that the probabilistic models tend to perform better than others and that combining different indexing and search strategies may further improve retrieval. The final performance is still lower than expected and further investigations are therefore needed in this domain.

1 Introduction

The Intellectual Property (IP) task is basically an *ad hoc* task with additional challenges. This year, the search consists of retrieving patents that are similar to a submitted one. This prior art search reflects the work of patent experts facing a newly submitted patent, having to decide if the submitted patent can be granted or not.

Contrary to other *ad hoc* retrieval tasks, we do not have a prescribed topic formulation but a whole patent document that can be used as a very long query. We think however that we are able to extract the most useful terms to formulate a more effective query. Furthermore we had to consider a vocabulary shift between the language used in the submitted patent and the language used in other patents. For example a patent proposal may not employ directly the word "pump" but may describe it with different words in order to avoid direct and evident conflict with existing patents. As an additional problem, the proposal may concern only a subpart of a given object (e.g., injection in a pump system) and pertinent items must be related not to the general object (e.g., pump) but to the specific targeted sub-component.

2 Overview of Test-Collection and IR Models

For the CLEF-IP track a corpus of 1,958,955 patent documents (XML format) was made available (covering the years 1985 to 2000). The patents provided by the European Patent Office (EPO) are written, in part, in English, German and French languages, with at least 100,000 documents in each language. Each patent can be divided into five main parts, namely "front page" (bibliography and

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 476–479, 2010.

abstract), state of the art, claims (actual protection), drawings with embedded examples, and finally the citations as well as search reports.

During the indexing, we decided to keep only following information: international patent classification number, abstract, the patent description, claims and the invention title. We also kept the language information in order to apply language specific indexing strategies (stemming, stopword or decomposition).

Based on our experiments, using the whole document as query was not an effective way to retrieved pertinent items. Thus we generated the query by applying the following procedure. For each term contained in the abstract, description, claim or invention title of the patent, we computed its *tf idf* weight. The m terms (fixed at 100 in our experiments) having the highest weights are selected to form the query. We reference to this query formulation as "Q". For some runs we additionally added the classification numbers contained in the patent (IPC codes). This second query formulation will be referenced as "QC".

In order to analyze the retrieval effectiveness under different conditions, we adopted various retrieval models for weighting the terms included in queries and documents. First we used a classical *tf idf* model with cosine normalization. We also implemented the $Lnu - ltc$ and $Lnc - ltc$ weighting schemes proposed by Buckley *et al.* [1]. To complete these vector-space models, we implemented several probabilistic approaches. As a first probabilistic approach, we implemented the Okapi model (BM25) as well as two models issued from the Divergence from Randomness (DFR) paradigm, namely $PL2$ and $InL2$. Finally we also used a statistical language model (LM) proposed by Hiemstra (for more details, see [2]).

During the indexing, we applied different strategies depending on the language. We eliminated very frequent terms using a language specific stopword list. Furthermore for each language the diacritics were replaced by their corresponding non-accented equivalent. We also applied a language-specific light stemming strategy. For the English language we used the S-Stemmer as proposed by Harman [3] and a stopword list containing 571 terms, while for the German language, we applied our light stemmer[1], a stopword list containing 603 words and a decompounding algorithm [4]. Finally for the French language we also used our light stemmer and a stopword list containing 484 words.

3 Evaluation

To measure the retrieval performance of our different runs, we adopted MAP values computed using the TREC_EVAL program. In the following tables, we have considered as relevant documents either "highly relevant" or simply "relevant".

Table 1 shows MAP achieved by 7 different IR models as well as the different indexing strategies and three query formulations. These evaluations were done using the small topic set (500). The last line depicts the MAP average over all IR models. To build the query, first we tried to weight search terms for each field (abstract, title, ...) separately and then added the results to obtain the final score. For example if one search term appears once in the abstract and

[1] http://www.unine.ch/info/clef/

once in the title, this term would have tf one for each field and a idf value related to the corresponding field before adding them to define the final weight. We reference to this strategy as "Separated Fields". Second we weighted the search terms considering the whole document, i.e., if a term t occurs once in two different fields it has tf of two. This strategy is denoted "Single Field". The third and last strategy consist in searching only the description of the patent ("Description"). Furthermore we applied two query formulations either taking into account classification numbers ("QC") or not ("Q").

As shown in Table 1, the language modeling approach (LM) tends to perform better than other IR models (highest MAP values depicted in bold). Using these best values as baseline, statistically significant performance differences are indicated by "†" (two-sided t-test, $\alpha = 5\%$). Based on this information, we can observe that in most cases the best performing model performs statistically better than the other search strategies. We can also observe that except if searching only in the description, vector-space models are generally outperformed by probabilistic models. Comparing the various query formulations, we can see that keeping the various fields separated (index "Separated Fields") shows slightly better performance. Finally, when searching only in the description field of the patent, we obtain similar performances as when searching in the whole patent document.

Table 1. MAP of Various IR Models and Query Formulations

	Mean Average Precision (MAP)			
Query	Q	QC	Q	Q
Index	Separated Fields	Separated Fields	Single Field	Description
Model / # of queries	500	500	500	500
Okapi	0.0832	0.0832 †	0.0843	0.0856 †
DFR-$InL2$	0.0849 †	0.0920 †	0.0645 †	0.0872 †
DFR-$PL2$	0.0830 †	0.0909 †	0.0515 †	0.0673 †
LM	**0.0886**	**0.0952**	**0.0891**	0.0787 †
$Lnc - ltc$	0.0735 †	0.0839 †	0.0554 †	**0.0884**
$Lnu - ltc$	0.0675 †	0.0782 †	0.0695 †	0.0589 †
$tf\ idf$	0.0423 †	0.0566 †	0.0380 †	0.0337 †
Average	0.0748	0.0829	0.0464	0.0714

Table 2 shows our official runs being either one single model or a fusion of several schemes based on the Z-score operator (more details are given in [2]). For the seven first strategies we used only the small topic set (500 queries) while for the last, we have used all 10,000 available topics. We observe that combining various runs tends to improve the retrieval effectiveness. The best performing strategy (UniNE_strat3) is a combination of two probabilistic models (DFR-$InL2$ and Okapi) and two different indexing strategies. The results for all runs lie relatively close together and present rather low MAP values. We do not consider expending automatically query formulation because the original topic

Table 2. Description and MAP of Our Official Runs

Run name	Query	Index	#Queries	Model	MAP	Comb.MAP
UniNE_strat1	Q	Single	500	$Lnu - ltc$	0.0695	0.0695
UniNE_strat2	Q	Single	500	LM	0.0891	0.0891
UniNE_strat3	QC	Separated	500	DFR-$InL2$	0.0920	**0.1024**
	Q	Description		Okapi	0.0856	
UniNE_strat4	Q	Separated	500	LM	0.0886	0.0961
	QC	Separated		Okapi	0.0832	
	Q	Single		LM	0.0891	
UniNE_strat5	Q	Description	500	Okapi	0.0856	0.0856
UniNE_strat6	QC	Separated	500	DFR-$PL2$	0.0909	0.0955
	Q	Single		$Lnu - ltc$	0.0554	
UniNE_strat7	QC	Separated	500	$Lnc - ltc$	0.0839	0.0839
UniNE_strat8	QC	Separated	10,000	Okapi	0.0994	0.0994

expression was already unusually long compared to other *ad hoc* tracks done in past CLEF evaluation campaigns.

4 Conclusion

We were not able to propose an effective procedure to extract the most useful search terms able to discriminate between the relevant and non-relevant patents. Our suggested selection procedure was based on *tf idf* weights and we experimented different indexing strategies and search models. It seems that building separate indexes for each field (title, description, abstract,...) and then combining the resulting ranked lists may improve the MAP. However, the resulting MAP values for almost all participating groups are below 0.1, indicating that further investigations are required in this domain.

Acknowledgments. This research was supported, in part, by the Swiss National Science Foundation under Grant #200021-113273.

References

1. Buckley, C., Singhal, A., Mitra, M., Salton, G.: New Retrieval Approaches Using SMART. In: Proceedings TREC-4, Gaithersburg, pp. 25–48 (1996)
2. Dolamic, L., Savoy, J.: Ad Hoc Retrieval with the Persian Language. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 102–109. Springer, Heidelberg (2010)
3. Harman, D.K.: How Effective is Suffixing? Journal of the American Society for Information Science 42, 7–15 (1991)
4. Savoy, J.: Report on CLEF-2003 Monolingual Tracks: Fusion of Probabilistic Models for Effective Monolingual Retrieval. In: Peters, C., Gonzalo, J., Braschler, M., Kluck, M. (eds.) CLEF 2003. LNCS, vol. 3237, pp. 322–336. Springer, Heidelberg (2004)

Automatically Generating Queries for Prior Art Search

Erik Graf, Leif Azzopardi, and Keith van Rijsbergen

University of Glasgow
{graf,leif,keith}@dcs.gla.ac.uk

Abstract. This paper outlines our participation in CLEF-IP's 2009 prior art search task. In the task's initial year our focus lay on the automatic generation of effective queries. To this aim we conducted a preliminary analysis of the distribution of terms common to topics and their relevant documents, with respect to term frequency and document frequency. Based on the results of this analysis we applied two methods to extract queries. Finally we tested the effectiveness of the generated queries on two state of the art retrieval models.

1 Introduction

The formulation of queries forms a crucial step in the workflow of many patent related retrieval tasks. This is specifically true within the process of Prior Art search, which forms the main task of the CLEF-IP 09 track. Performed both, by applicants and the examiners at patent offices, it is one of the most common search types in the patent domain, and a fundamental element of the patent system. The goal of such a search lies in determining the patentability (See Section B IV 1/1.1 in [3] for a more detailed coverage of this criterion in the European patent system) of an application by uncovering relevant material published prior to the filing date of the application. Such material may then be used to limit the scope of patentability or completely deny the inherent claim of novelty of an invention. As a consequence of the judicial and economic consequences linked to the obtained results, and the complex technical nature of the content, the formulation of prior art search queries requires extensive effort. The state of the art approach consists of laborious manual construction of queries, and commonly requires several days of work dedicated to the manual identification of effective keywords. The great amount of manual effort, in conjunction with the importance of such a search, forms a strong motivation for the exploration of techniques aimed at the automatic extraction of viable query terms. Throughout the remainder of these working notes we will provide details of the approach we have taken to address this challenge. In the subsequent section we will provide an overview of prior research related to the task of prior art search. Section 3 covers the details of our experimental setup. In section 4 we report on the official results and perform an analysis. Finally in the last section we provide a conclusion and future outlook.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 480–490, 2010.

2 Prior Research

The remainder of this section aims at providing an overview of prior research concerning retrieval tasks related to the CLEF-IP 09 task. As the majority of relevant retrieval research in the patent domain has been pioneered by the NTCIR series of evaluation workshops [1], additionally a brief overview of relevant collections and the associated tasks is provided. Further we review a variety of successful techniques applied by participating groups of relevant NTCIR tasks.

First introduced in the third NTCIR workshop [9], the patent task has led to the release of several patent test collections. Details of these collections are provided in Table 1. From the listing in Table 1 we can see that the utilized collections are comparative in size to the CLEF-IP 09 collection, and that the main differences consist of a more limited time period and a much smaller amount of topics specifically for the earlier collections.

Table 1. Overview of NTCIR patent test collections (E=English, J=Japanese)

Workshop	Document Type	Time Period	# of Docs.	# of Topics
NTCIR-3	Patent JPO(J)	1998-1999	697,262	31
	Abstracts(E/J)	1995-1999	ca. 1,7 million	31
NTCIR-4	Patent JPO(J), Abstracts(E)	1993-1997	1,700,000	103
NTCIR-5	Patent JPO(J), Abstracts(E)	1993-2002	3,496,252	1223
NTCIR-6	Patent USPTO(E)	1993-2002	1,315,470	3221

Based on these collections the NTCIR patent track has covered a variety of different tasks, ranging from cross-language and cross-genre retrieval (NTCIR 3 [9]) to patent classification (NTCIR 5 [7] and 6 [8]). A task related to the Prior Art search task is presented by the invalidity search run at NTCIR 4 [4],5 [5], and 6 [6]). Invalidity searches are exercised in order to render specific claims of a patent, or the complete patent itself, invalid by identifying relevant prior art published before the filing date of the patent in question. As such, this kind of search, that can be utilized as a means of defense upon being charged with infringement, is related to prior art search. Likewise the starting point of the task is given by a patent document, and a viable corpus may consist of a collection of patent documents. In course of the NTCIR evaluations, for each search topic (i.e. a claim), participants were required to submit a list of retrieved patents and passages associated with the topic. Relevant matter was defined as patents that can invalidate a topic claim by themselves (1), or in combination with other patents (2). In light of these similarities, the following listing provides a brief overview of techniques applied by participating groups of the invalidity task at NTCIR 4-6:

– **Claim Structure Based Techniques:** Since the underlying topic consisted of the text of a claim, the analysis of its structure has been one of the commonly applied techniques. More precisely the differentiation between premise

and invention parts of a claim and the application of term weighting methods with respect to these parts has been shown to yield successful results.

- **Document Section Analysis Based Techniques:** Further one of the effectively applied assumptions has been, that certain sections of a patent document are more likely to contain useful query terms. For example it has been shown that from the 'detailed descriptions corresponding to the input claims, effective and concrete query terms can be extracted' NTCIR 4 [4].
- **Merged Passage and Document Scoring Based Techniques:** Further grounded on the comparatively long length of patent documents, the assumption was formed that the occurrence of query terms in close vicinity can be interpreted as a stronger indicator of relevance. Based on this insight, a technique based on merging passage and document scores has been successfully introduced.
- **Bibliographical Data Based Techniques:** Finally the usage of bibliographical data associated with a patent document has been applied both for filtering and re-ranking of retrieved documents. Particularly the usage of the hierarchical structure of the IPC classes and applicant identities have been shown to be extremely helpful. The NTCIR 5 proceedings [5] cover the effect of applying this technique in great detail and note that, 'by comparing the MAP values of Same' (where Same denotes the same IPC class) 'and Diff in either of Applicant or IPC, one can see that for each run the MAP for Same is significantly greater than the MAP for Diff. This suggests that to evaluate contributions of methods which do not use applicant and IPC information, the cases of Diff need to be further investigated.' [5]. The great effectiveness is illustrated by the fact that for the mandatory runs of NTCIR the best reported MAP score for 'Same' was 0,3342 MAP whereas the best score for 'Diff' was 0,916 MAP.

As stated before our experiments focused on devising a methodology for the identification of effective query terms. Therefore in this initial participation, we did not integrate the above mentioned techniques in our approach. In the following section the experimental setup and details of the applied query extraction process will be supplied.

3 Experimental Setup

The corpus of the CLEF-IP track consists of 1,9 million patent documents published by the European Patent Office (EPO). This corresponds to approximately 1 million individual patents filed between 1985 and 2000. As a consequence of the statutes of the EPO, the documents of the collection are written in English, French and German. While most of the early published patent documents are mono-lingual, most documents published after 2000 feature title, claim, and abstract sections in each of these three languages. The underlying document format is based on an innovative XML schema[1] developed at Matrixware[2].

[1] http://www.ir-facility.org/pdf/clef/patent-document.dtd
[2] http://www.matrixware.com/

Indexing of the collection took place utilizing the Indri[3] and Lemur retrieval system[4]. To this purpose the collection was wrapped in TREC format. The table below provides details of the created indices:

Table 2. Clef-IP 09 collection based indices

Index Name	Retrieval System	Stemming	Stop-Worded	UTF-8
Lem-Stop	Lemur	none	Stop-worded	No
Indri-Stop	Indri	none	Stop-worded	Yes

As can be seen from the table we did not apply any form of stemming on both indices. This decision was based on the fact that the corpus contains a large amount of technical terms (e.g. chemical formulas) and tri-lingual documents. In order to increase indexing efficiency, stop-wording based on the English language was applied to all indices. A minimalistic stop-word list was applied in order to mitigate potential side effects. The challenges associated with stop-wording in the patent domain are described in more detail by Blanchard [2]. No stop-wording for French and German was performed. The creation of the Indri-Stop index was made necessary in order to allow for experiments based on the filtering terms by language. Lemur based indices do not support UTF-8 encoding and therefore did not allow for filtering of German or French terms by use of constructed dictionaries.

3.1 Effective Query Term Identification

As stated before the main aim of our approach lies in the extraction of effective query terms from a given patent document. The underlying assumption of our subsequently described method is, that such terms can be extracted based on an analysis of the distribution of terms common to a patent application and its referenced prior art.

The task of query extraction therefore took place in two phases: In the first phase we contrasted the distribution of terms shared by source documents and referenced documents with the distribution of terms shared by randomly chosen patent document pairs. Based on these results the second phase consisted of the extraction of queries and their evaluation based on the available CLEF-IP 09 training data. In the following subsections both steps are discussed in more detail.

3.1.1 Analysing the common term distribution. The main aim of this phase lies in the identification of term related features whose distribution varies among source-reference pairs and randomly chosen pairs of patent documents. As stated before the underlying assumption is, that such variations can be utilized

[3] http://www.lemurproject.org/indri/
[4] http://www.lemurproject.org/lemur/

in order to extract query terms whose occurrences are characteristic for relevant document pairs. To this extent we evaluated the distribution of the following features:

1. The corpus wide term frequency (tf)
2. The corpus wide document frequency (df)

In order to uncover such variations the following procedure was applied: For a given number of n source-reference pairs an equal number of randomly chosen document pairs was generated. Secondly the terms common to document pairs in both groups were identified. Finally an analysis with respect to the above listed features was conducted.

As a result of this approach figure 1 depicts the number of common terms for source-reference pairs and randomly chosen pairs with respect to the corpus wide term frequency. In the graph, the x-axis denotes the collection wide term frequency, while on the y-axis the total number of occurrences of common terms with respect to this frequency is provided. Evident from the graph are several high-level distinctive variations: The first thing that can be observed is that the total number of shared terms of source-reference pairs is higher than for those of random pairs. Further the distribution of shared terms in random pairs, shown in blue, resembles a straight line on the log-log scale. Assuming that the distribution of terms in patent documents follows a Zipf like distribution this can be interpreted as an expected outcome. In contrast to this, the distribution of shared terms in source-reference pairs, depicted in red, varies significantly. This is most evident in the low frequency range of approximately 2-10000.

Given our initial goal of identifying characteristic differences in the distribution of terms shared within relevant pairs, this distinctive pattern can be utilized as a starting point of the query extraction process. Therefore, as will be evident in more detail in the subsequent section, we based our query extraction process on this observation.

3.2 Query Extraction

Based on the characteristic variations of the distribution of terms common to source-reference pairs our query term extraction process uses the document frequency as selection criterion. The applied process hereby consisted of two main steps. Based on the identification of the very low document frequency range as most characteristic for source-reference pairs, we created sets of queries with respect to the df of terms (1) for each training topic. These queries were then submitted to a retrieval model, and their performance was evaluated by use of the available training relevance assessments (2).

Following this approach two series of potential queries were created via the introduction of two thresholds.

– **Document Frequency (df) Based Percentage Threshold:** Based on this threshold, queries are generated by including only terms whose df lies below an incrementally increased limit. To allow for easier interpretation the

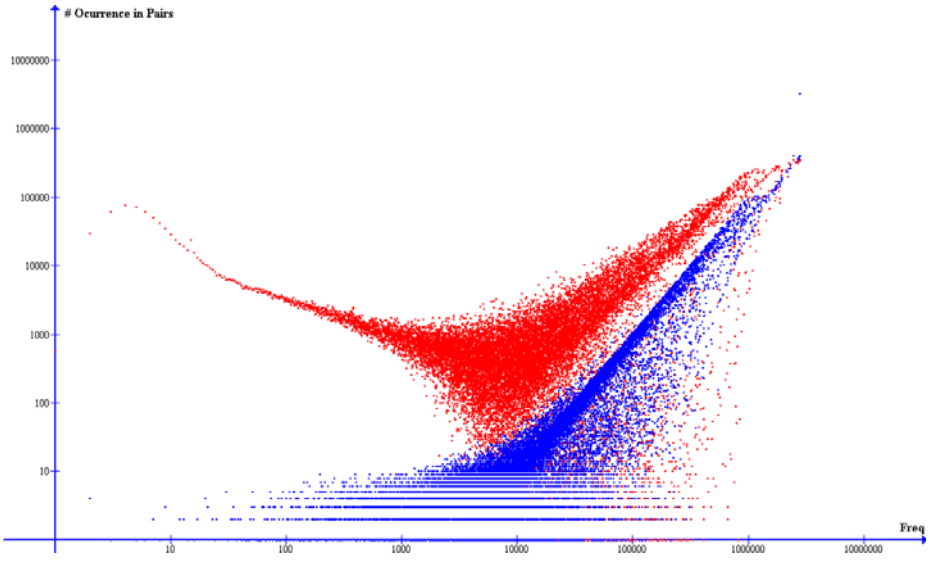


Fig. 1. Distribution of shared terms: Source-Reference pairs versus Random Pairs

incrementally increased limit is expressed as $\frac{df}{N} * 100$, where N denotes the total number of documents in the collection. A percentage threshold of 0.5% therefore denotes, that we include only terms in the query that appear in less than 0.5% of the documents in the collection.

- **Query Length Threshold:** A second set of queries was created by utilization of an incrementally increased query length as underlying threshold. In this case for a given maximum query length n, a query was generated by including the n terms with the lowest document frequency present in the topic document. The introduction of this threshold was triggered by the observation that the amount of term occurrences with very low df varies significantly for the topic documents. As a consequence of this a low df threshold of 1000 can yield a lot of query terms for some topics, and in the extreme case no query terms for other topics.

We generated queries based on a percentage threshold ranging from 0.25%-3% with an increment of 0.25, and with respect to query lengths ranging from 10-300 with an increment of 10. The performance of both query sets was then evaluated by utilization of the large training set of the main task with the BM25 and Cosine retrieval models.

Figure 2 depicts the MAP and Recall scores based on a series of df-threshold based queries using the BM25 retrieval model. Scores for the Cosine model based on varying query length are shown in Figure 3.

The first thing we observed from these training topic runs, is that the applied methods of query formulation return promising results for both retrieval models, and that this is the case for both query extraction strategies. Secondly, BM25

always exhibited a higher performance with respect to both MAP and the number of retrieved documents. The higher slope of the graph showing the performance of the cosine retrieval model is not mainly induced by the properties of the model itself, but rather through the length of the applied queries. The average query length for a percentage threshold of 0.25 (the first data point) for example was 198.276. By applying lower df thresholds, which would result in considerably shorter queries, a similar graph can be witnessed for the performance of BM25. During our training phase the percentage-threshold method showed slightly better results. We believe that a possible explanation may consist of an increased potential for topic drift that can be introduced by allowing for the inclusion of terms with higher df for large query length thresholds.

4 Results and Analysis

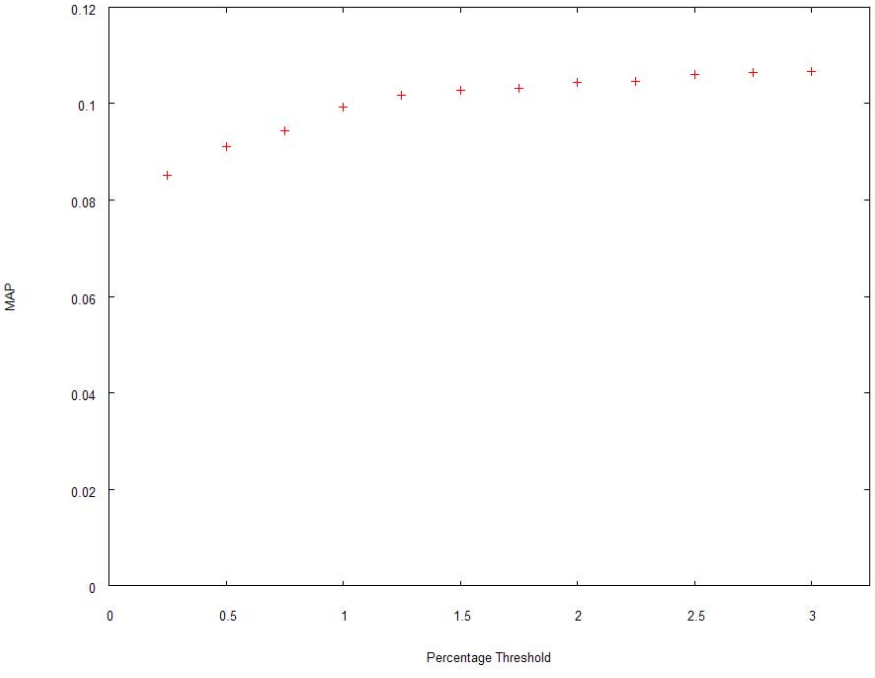
In the following a description of the submitted runs and an analysis of their performance will be conducted. In total our group submitted five runs. While the performance of one of our runs is in line the with the observations based on the training set, the performance of the other four runs resulted in a completely different and order of magnitudes lower results. Unfortunately this was induced by a bug occurring in the particular retrieval setup utilized for their creation. The amount of analysis that can be drawn from the official results is therefore very limited. After obtaining the official qrels we re-evaluated the baseline run of these four runs in order to verify the observed tendencies of the training phase.

4.1 Description of Submitted Runs and Results

We participated in the Main task of this track with four runs for the Medium set that contains 1,000 topics in different languages. All four runs where based on the BM25 retrieval model using standard parameter values (b = 0.75, k1 = 1.2, k3 =1000), and utilized a percentage threshold of 3.0. These runs are listed below:

- BM25medStandard: No filtering of query terms by language was applied. Query terms where selected solely considering their df.
- BM25EnglishTerms: German and French terms were filtered out.
- BM25FrenchTerms: English and German terms were filtered out.
- BM25GermanTerms: English and French terms were filtered out.

Additionally we submitted a run for the XL set consisting of 10000 topics. This run also utilized a threshold of 3.0, used the Cosine retrieval model, and filtered out French and German terms via the utilization of dictionaries that were constructed based on the documents in the Clef-IP 09 corpus. Table 4 lists the official results of the above described runs.



(a) MAP for varying percentage thresholds with the BM25 Model

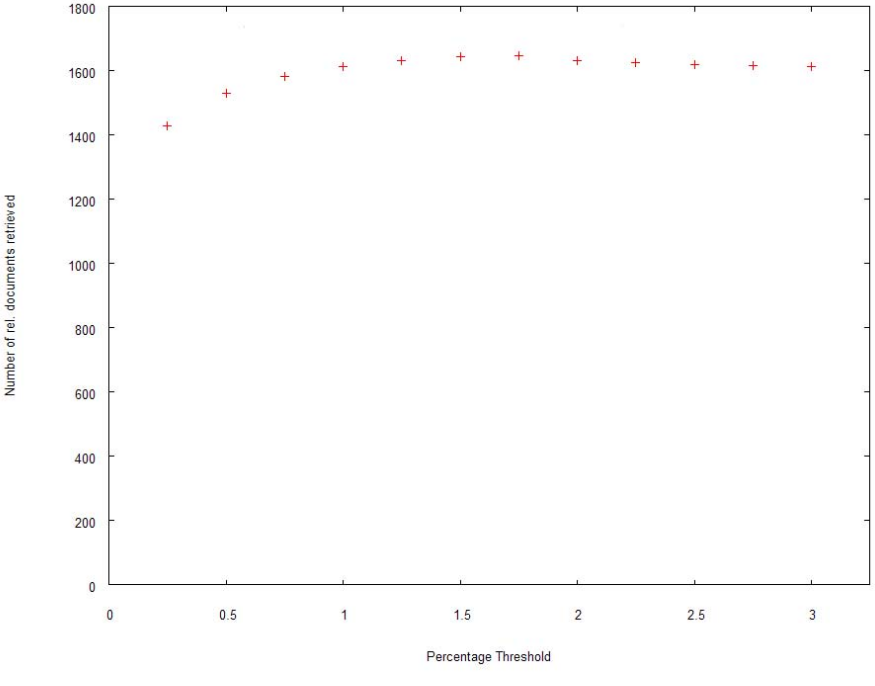
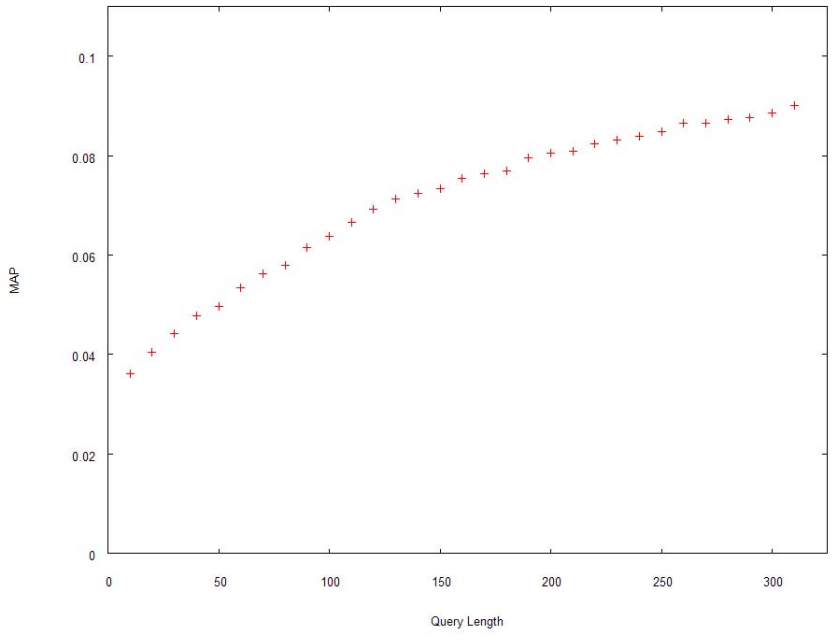
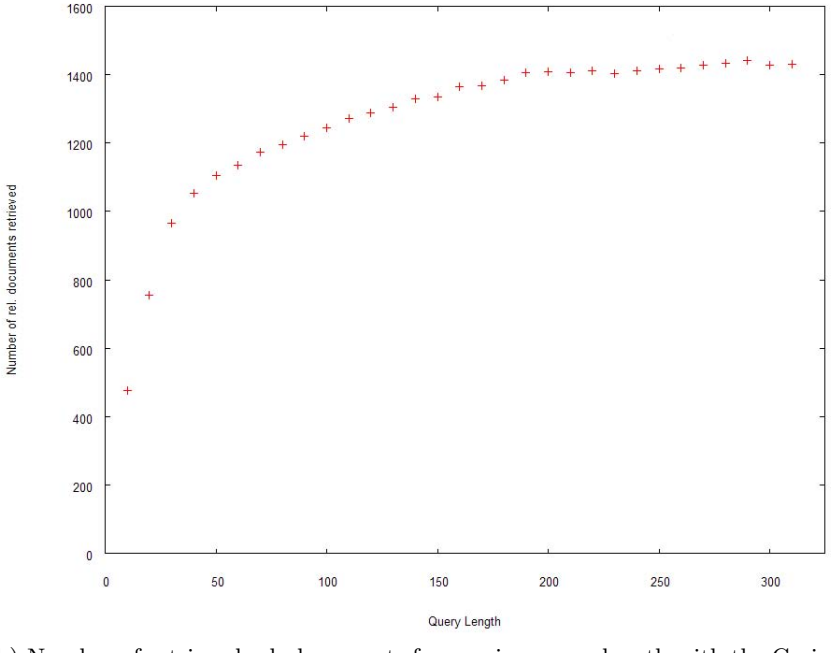


Fig. 2. Results for varying percentage thresholds for the BM25 model



(a) MAP performance for varying query length with the Cosine model



(b) Number of retrieved rel. documents for varying query length with the Cosine model

Fig. 3. Results for varying query length for the cosine model

Table 3. Official Run Results

run id	P	P5	P10	P100	R	R5	R10	R100	MAP	nDCG
BM25medstandard	0.0002	0.0000	0.0001	0.0002	0.0238	0.0000	0.0001	0.0033	0.0002	0.0389
BM25EnglishTerms	0.0001	0.0002	0.0001	0.0001	0.0159	0.0002	0.0002	0.0017	0.0002	0.0318
BM25FrenchTerms	0.0001	0.0004	0.0002	0.0002	0.0123	0.0004	0.0004	0.0027	0.0003	0.0270
BM25GermanTerms	0.0001	0.0002	0.0001	0.0001	0.0159	0.0002	0.0002	0.0017	0.0002	0.0318
CosEnglishTerms	0.0036	0.0854	0.0600	0.0155	0.4667	0.0808	0.1100	0.2599	0.0767	0.4150

4.2 Analysis

While the CosEnglishTerms run showed comparable performance to the observations during the training phase outlined in Figure 3, it can be seen from the results that the performance of the BM25 based runs was significantly lower than the observed results in Figure 2 . Therefore first of all, it is not possible for us to draw any conclusions towards the effect of the applied filtering by language from these results. In retrospective analysis we identified that this almost complete failure in terms of performance was linked to applying the BM25 model to the Indri indices created to allow for language filtering. While this problem has not yet been resolved and we were therefore not able to re-evaluate the language filtering based runs, we re-evaluated the BM25medstandard run using a Lemur based index and the released official qrels. This resulted in the below listed performance, that is in the same range of what we witnessed during the BM25 training phase. It confirms our observation that BM25 seems to be more effective than the Cosine model.

Table 4. Re-evaluated run result

run id	P5	P10	P100	R	MAP
BM25medstandard	0.1248	0.0836	0.0188	0.511	0.1064

5 Conclusion and Future Outlook

Based on one of the submitted runs and our training results this preliminary set of experiments has shown that our proposed method of automatic query formulation may be interpreted as a promising start towards effective automatic query formulation. As such a technique may significantly facilitate the process of prior art search through the automatic suggestion of efficient keywords, it is planned to extend our experimentation in several directions. These extensions include the consideration of a patent document's structure (i.e. title, description, claims) in the selection process, and the introduction of a mechanism that will allow the weighted inclusion of term related features in addition to the document frequency.

Acknowledgements

We would like to express our gratitude for being allowed to use the Large Data Collider (LDC) computing resource provided and supported by the Information Retrieval Facility[5]. We specifically also want to thank Matrixware[6] for having co-funded this research.

References

[1] National institue of informatics test collection for ir systems (ntcir), http://research.nii.ac.jp/ntcir/
[2] Blanchard, A.: Understanding and customizing stopword lists for enhanced patent mapping. World Patent Information 29(4), 308–316 (2007)
[3] European Patent Office (EPO). Guidelines for Examination in the European Patent Office (December 2007)
[4] Fujii, A., Iwayama, M., Kando, N.: Overview of patent retrieval task at ntcir-4. In: Proceedings of NTCIR-4 Workshop Meeting (2004)
[5] Fujii, A., Iwayama, M., Kando, N.: Overview of patent retrieval task at ntcir-5. In: Proceedings of NTCIR-5 Workshop Meeting (2005)
[6] Fujii, A., Iwayama, M., Kando, N.: Overview of the patent retrieval task at the ntcir-6 workshop. In: Proceedings of NTCIR-6 Workshop Meeting, pp. 359–365 (2007)
[7] Iwayama, M., Fujii, A., Kando, N.: Overview of classification subtask at ntcir-5 patent retrieval task. In: Proceedings of NTCIR-5 Workshop Meeting (2005)
[8] Iwayama, M., Fujii, A., Kando, N.: Overview of classification subtask at ntcir-6 patent retrieval task. In: Proceedings of NTCIR-6 Workshop Meeting, pp. 366–372 (2007)
[9] Iwayama, M., Fujii, A., Kando, N., Takano, A.: Overview of patent retrieval task at ntcir-3. In: Proceedings of NTCIR-3 Workshop Meeting (2002)

[5] http://www.ir-facility.org/
[6] http://www.matrixware.com/

Patent Retrieval Experiments in the Context of the CLEF IP Track 2009

Daniela Becks, Christa Womser-Hacker, Thomas Mandl, and Ralph Kölle

Information Science, University of Hildesheim, Germany
{Daniela.Becks,womser,mandl,koelle}@uni-hildesheim.de

Abstract. At CLEF 2009 the University of Hildesheim submitted experiments for the new Intellectual Property Track. We focused on the main task of this track that aims at finding prior art for a specified patent. Our experiments were split up into one official German run as well as different additional runs using English and German terms. The submitted run was based on a simple baseline approach including stopword elimination, stemming and simple term queries. Furthermore, we investigated the significance of the International Patent Classification (IPC). During the experiments, different parts of a patent were used to construct the queries. In a first stage, only title and claims were included. In contrast, for the post runs we generated a more complex boolean query, which combined terms of the title, claims, description and the IPC classes. The results made clear that using the IPC codes can particularly increase the recall of a patent retrieval system.

1 Introduction

Since there is a growing number of patent applications, the importance of the Intellectual Property domain is increasing [1]. This tendency can be seen not just in industrial contexts but also in academia. Particularly in Information Science, where information seeking and behaviour subjects are of main interest, patent documents play a vital role because they bear a lot of differences especially at the terminology level [2] [3]. Therefore, one of the most challenging tasks to information scientists is to adapt a retrieval system to these specialties. This, of course, raises the question of how to deal with the characteristics of the patent domain.

In 2009 the Cross Language Evaluation Forum offered a special track, the Intellectual Property Track, whose aim is to apply different information retrieval techniques to this specific domain [4]. The main task of this track focused on prior art search that is performed to determine whether an invention or part of it already exists [2]. Any document that states prior art is relevant to the query.

1.1 Test Collection and Topics

The test collection consisted of about 1.9 million patent documents of the European Patent Office, which have been stored as XML documents [4]. A patent

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 491–496, 2010.

may be formulated in German, English or French, whereas the English language appeared most frequently within the collection [4]. Furthermore, different topic sets, ranging from 500 to 10.000 patent documents, were provided by the organizers [4]. Our experiments concentrated on the smallest topic set.

The documents of the test collection as well as the topic files use the typical patent structure. In other words, they are divided into the following sections [2]:

- Bibliographic data (e.g. name of the inventor, patent number)
- Disclosure (e.g. title, abstract and detailed description)
- Claims

The beginning of each patent contains meta information like the name of the inventor. Furthermore, the second passage provides a brief and a detailed description of the invention. The third section comprises a number of different claims, one of which is the main claim. Besides this information, a patent comprehends several classification codes depending on the patent classification used. In case of the test collection, the documents contain IPC[1] and ECLA[2] codes. With respect to information retrieval, the second and third section are of major interest because they contain patent specific vocabulary, which will be discussed in the next section (Sect. 1.2).

1.2 Patent Terminology

Due to their terminology and structure, patent documents differ significantly from other types of documents. Many scientists have already dealt with these differences.

According to [5], there exist a couple of laws describing formal requirements that a applicant has to consider when writing a patent. [6] confirmed this fact. These regulations not only influence the structure but also the terminology of the patents. Therefore, certain expressions are likely to appear more often.

The argument mentioned above particularly refers to claims, which in German have to contain phrases like "dadurch gekennzeichnet, dass" [7]. The same applies to words like "comprises", "claim" or the German equivalents "umfasst" and "Anspruch".

Another characteristic of patent documents is the rather general and vague vocabulary used by many applicants [2]. In the test collection as well as in the topic set this is the case for terms and phrases like "system", "apparatus", "method" or "Verfahren und Vorrichtung zur" (e.g. Patent number EP-1117189).

Keeping this fact in mind, a simple tf-idf approach is critical because such terms appear frequently in the patent documents. As a consequence, the result list returned by the system is supposed to be quite comprehensive and precision is going to decrease.

Furthermore, patent documents contain a huge amount of technical terms [2]. In the dataset, we figured out different types of technical vocabulary.

[1] International Patent Classification.
[2] European Patent Classification.

- compounds like *Dieselbrennkraftmaschinen* (Patent number EP-1114924)
- hyphenated words, e.g. *AGR-System* (Patent number EP-1114924)
- acronyms like *EGR* (Patent number EP-1114924)
- physical metrics like *1.2288 MHz* (Patent number EP-1117189)
- numbers
- chemical symbols, e.g. *Cis-Diamminoplatin(II)chlorid* (Patent number EP-1520870)

Technical terms of this type often appear in the claims and require well-adapted parsing and stemming techniques. A specific problem that one might come across with is hyphenation because removing the hyphen would split up a compound into single words with different meanings. This might also lead to quite low precision.

2 System Setup

Our experiments were performed using a simple retrieval system based on Apache Lucene[3]. Although Lucene allows the use of boolean logic, it is mainly based on the well-known Vector Space Model. Thus, the ranking algorithm is a simple tf-idf approach. The following section provides a description of our approach.

2.1 Preprocessing and Indexing

We restricted our experiments to monolingual runs based on an English and a German index.

As the documents existed in XML format, we first had to extract the content from the fields that should be indexed whereas for the English index only the text written in English has been considered. The same procedure was used for building up a German index file.

In the case of our submitted German run, we followed the argumentation that the claims are supposed to be most important during prior art searches [2] and stored the text of this section into the German index. We further added the title and the patent number. While the last one only served as an identifier for the document and has not been stemmed, the German Analyzer, a stemmer specifically provided for the German language, was applied to the content of the title and the claims. For the post runs we created a more comprehensive English and German index by adding the IPC codes as well as the detailed description found in patent documents. The fields mentioned above (claims, title, patent number) were kept, but were stemmed with the Snowball Stemmer[4] instead.

As we performed monolingual runs with English and German we integrated one specific stopword list for each language. As described in Sect. 1.2, in the intellectual property domain some terms are likely to appear quite often and would return a huge amount of results. To solve this problem, we enriched the

[3] http://lucene.apache.org/

[4] http://snowball.tartarus.org/

German and English stopword lists[5] with some patent specific and a couple of vague terms like *apparatus*, *method*, *comprising* or *claim*.

2.2 Search Process

The main task of the track focused on prior art search that is performed to determine whether an invention or part of it already exists [2]. Any document that states prior art is relevant to the query.

In the context of the Intellectual Property Track a topic is a given patent. Following this, queries are generated on the basis of the terms extracted from the topic files, which have been preprocessed similarly to the documents of the test collection.

During the experiments we tested out different query types:

1. simple term queries without any boolean operator (official run)
2. boolean queries combining IPC codes and terms of the claims/ all fields (post runs, see Sect. 3.2)

3 Results and Analysis

The experiments of the University of Hildesheim concentrated on the main task of the Intellectual Property Track 2009.

3.1 Submitted Run

We submitted one German run within the main task. Therefore, simple term queries were conducted on the German index originally consisting of three fields (*UCID*, *CLAIM-TEXT* and *INVENTION-TITLE*). Only the title and the claims were used to generate the queries and documents were ranked according to tf-idf. More information can be found in [8].

The results of our official run did not satisfy our expectations. Recall and precision were close to zero. In other words, our system returned a lot of documents, wherein most of them did not state prior art of the query patent.

3.2 Post Runs

Besides the submitted run mentioned in the previous section, we ran further experiments to develop a more sophisticated search strategy.

The results of the German official run imply that simple term queries may not be the best solution to perform prior art searches. We therefore decided to construct a more complex boolean query combining IPC codes and terms extracted from different sections of the topic files.

We assumed that IPC codes could be useful to filter the initial document set. As a consequence, we included the classification information into the boolean

[5] http://members.unine.ch/jacques.savoy/clef/index.html

query. A patent document that is relevant to the query is supposed to share at least one IPC code or, in case of the runs using all fields, the most frequent one with the topic from which the query was generated. Additionally, we extracted the content of the claims, the title and the description of the topic files.

The obtained search terms were combined using the *OR* operator and integrated into the boolean query.

The post runs and the employed settings are listed below:

1. **EN_Snow_allFields**: EN index, Snowball, IPC (most frequent),title, claims, description
2. **DE_Snow_allFields**: DE index, Snowball, IPC (most frequent),title, claims, description
3. **EN_Snow_TitleMainclaim**: EN index, Snowball, IPC (at least one), title and first claim
4. **EN_Snow_TitleClaim20**: EN index, Snowball, IPC (at least one), title and 20 terms of first claim

Table 1 provides some statistics according to the obtained results, which were calculated utilizing the latest version, which is version 9.0, of trec_eval[6].

Table 1. Evaluation measures

	Recall	**Precision**	**MAP**
official run	**0.0487**	**0.0019**	**0.0031**
DE_Snow_allFields	0.2042	0.0025	0.0160
EN_Snow_allFields	0.2052	0.0026	0.0200
EN_Snow_TitleMainclaim	0.4392	0.0057	0.0488
EN_Snow_TitleClaim20	0.4202	0.0054	0.0491

A look at the results shown in table 1 reveals that integrating the IPC codes improved recall. In each case of the post runs, we achieved a recall of at least 0.20. For the runs using all IPC codes, the recall was even higher (about 0.43). In contrast, the system submitted to the official run only retrieved 5% of the relevant documents. We also managed to increase the mean average precision (MAP) of our retrieval system. As it can be seen, the run based on the English index achieved a better MAP (0.02) than the German one (0.016) and the runs without using the description even performed better. Still, a MAP of about 0.05 longs for further experiments.

Another important fact that can be figured out is that the query seems to be most precise if terms extracted from the title and the claims are included. The results of the run utilizing the claims only and the runs combining all fields did not achieve the same MAP as the runs using both title and claims. We supposed that we should restrict the number of terms the query consists of. Therefore, we

[6] http://trec.nist.gov/trec_eval/

extracted only terms from the first claim. The initial hypothesis that a cutoff could increase precision could not be confirmed. As it can be seen, there is hardly a difference between the results of the run based on the complete first claim and the run, which has been performed with a cutoff of 20 query terms extracted from the introducing claim.

4 Outlook

The patent retrieval domain is quite different from other ones. Especially the linguistic features like terminology and text structure bear a lot of difficulties. Through some additional experiments we were able to figure out that taking into account the IPC codes of the patent documents can improve the recall of a patent retrieval system. Furthermore, we would recommend a combination of terms extracted from title and claims. Restricting the number of query terms to 20 did not achieve a significant difference wrt precision.

In the future, we are planning to implement a more sophisticated search strategy including e.g. *Query Expansion*. We are also thinking about applying NLP techniques during preprocessing because by now, we have implemented a baseline approach without regarding the patent specific structure and terminology.

References

1. European Patent Office: The economic importance of patents. Rich in intellectual property (2008), http://www.epo.org/topics/innovation-and-economy/economic-impact.html (verified: 27.02.2010)
2. Graf, E., Azzopardi, L.: A methodology for building a test collection for prior art search. In: Proceedings of the 2nd International Workshop on Evaluating Information Access (EVIA), Tokyo, Japan, December 16, pp. 60–71 (2008)
3. Kando, N.: What Shall We Evaluate? - Preliminary Discussion for the NTCIR Patent IR Challenge (PIC) Based on the Brainstorming with the Specialized Intermediaries in Patent Searching and Patent Attorneys. In: ACM-SIGIR Workshop on Patent Retrieval, Athens, Greece, July 28, pp. 37–42 (2000)
4. Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: retrieval experiments in the Intellectual Property domain. In: CLEF Working Notes 2009, Corfu, Greece (2009)
5. Schamlu, M.: Patentschriften - Patentwesen. Eine argumentationstheoretische Analyse der Textsorte Patent am Beispiel der Patentschriften zu Lehrmitteln. Indicium-Verlag, München (1985)
6. Ahmad, K., Al-Thubaity, A.: Can Text Analysis Tell us Something about Technology Progress? In: Proceedings of the ACL-2003 Workshop On Patent Corpus Processing, Sapporo, Japan, July 12 (2003)
7. Deutsches Patent- und Markenamt: Merkblatt für Patentanmelder (2009), http://www.dpma.de/docs/service/formulare/patent/p2791.pdf
8. Becks, D., Womser-Hacker, C., Mandl, T., Kölle, R.: Patent Retrieval Experiments in the Context of the CLEF IP Track 2009. In: CLEF Working Notes 2009, Corfu, Greece (2009)

Prior Art Retrieval Using the Claims Section as a Bag of Words

Suzan Verberne and Eva D'hondt

Information Foraging Lab, Radboud University Nijmegen
{s.verberne,e.dhondt}@let.ru.nl

Abstract. In this paper we describe our participation in the 2009 CLEF-IP task, which was targeted at prior-art search for topic patent documents. We opted for a baseline approach to get a feeling for the specifics of the task and the documents used. Our system retrieved patent documents based on a standard bag-of-words approach for both the Main Task and the English Task. In both runs, we extracted the claim sections from all English patents in the corpus and saved them in the Lemur index format with the patent IDs as DOCIDs. These claims were then indexed using Lemur's BuildIndex function. In the topic documents we also focused exclusively on the claims sections. These were extracted and converted to queries by removing stopwords and punctuation. We did not perform any term selection or query expansion. We retrieved 100 patents per topic using Lemur's RetEval function, retrieval model TF-IDF. Compared to the other runs submitted to the track, we obtained good results in terms of nDCG (0.46) and moderate results in terms of MAP (0.054).

1 Introduction

In 2009 the first CLEF-IP track was launched by the Information Retrieval Facility (IRF)[1] as part of the CLEF 2009 evaluation campaign.[2] The general aim of the track was to explore patent searching as an IR task and to try to bridge the gap between the IR community and the world of professional patent search.

The goal of the 2009 CLEF-IP track was "to find patent documents that constitute prior art[3] to a given patent" [1]. In this retrieval task each topic query was a (partial) patent document which could be used as one long query or from which smaller queries could be generated. The track featured two kinds of tasks: In the Main Task prior art had to be found in any one (or combination) of the three following languages: English, French and German; three optional subtasks used parallel monolingual topics in one of the three languages. In total 15 European teams participated in the track.

[1] See http://www.ir-facility.org/the_irf/clef-ip09-track

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

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

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 497–501, 2010.

At the Radboud University of Nijmegen we decided to participate in the CLEF-IP track because it is related to the focus of the Text Mining for Intellectual Property (TM4IP) project[4] that we are currently carrying out. In this project we investigate how linguistic knowledge can be used effectively to improve the retrieval process and facilitate interactive search for patent retrieval. Because the task of prior-art retrieval was new to us, we chose to implement a baseline approach to investigate how well traditional IR techniques work for this type of data and where improvements would be most effective. These results will effectively serve as a baseline for further experiments as we explore the influence of using dependency triplets[5] for various IR tasks on the same patent corpus.

2 Our Methodology

2.1 Data Selection

The CLEF-IP corpus consists of EPO documents with publication date between 1985 and 2000, covering English, French, and German patents (1,958,955 patent-documents pertaining to 1,022,388 patents, 75GB) [2]. The XML documents in the corpus do not correspond to one complete patent each but one patent can consist of multiple XML files (representing documents that were produced at different stages of a patent realization).

In the CLEF-IP 2009 track, the participating teams were provided with 4 different sets of topics (S,M,L,XL). We opted to do runs on the smallest set (the S data set) for both the Main and the English task. This set contained 500 topics. There appeared to be a number of differences in the information that is contained in the topics for the Main task and the English task: the topics for the Main Task contained the abstract content as well as the full information of the granted patent except for citation information, while the topic patents for the English Task only contained the title and claims sections of the granted patent [2].

Therefore, we decided to use the field that was available in all topics for both tasks: the (English) claims sections. Moreover, as [3], [4] and [5] suggest, the claims section is the most informative part of a patent, at least for prior-art search. We found that 70% of the CLEF-IP corpus contained English claims, as a result of which a substantial part of the corpus was excluded from our experiments. Of the 30% that could not be retrieved by our system, 7% were documents that only had claims in German or French but not in English, 6% only contained a title and abstract, usually in English and 17% only contained a title.

2.2 Query Formulation

At the CLEF-IP meeting there was much interest on term extraction and query formulation, for example from the University of Glasgow [6]. Though this seems

[4] http://www.phasar.cs.ru.nl/TM4IP.html

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

to be a promising topic, we choose not to distil any query terms from the claims sections but instead concatenated all words in the claims section in one long query. The reason for this was twofold. First, adding a term selection step in the retrieval process makes the retrieval process more prone to errors because it requires the development of a smart selection process. Second, by weighting the query and document terms using the TF-IDF ranking model, a form of term selection is carried out in the retrieval and ranking process. We did not try to enlarge the set of query words with any query expansion technique but only used the words as they occurred in the texts.

2.3 Indexing and Retrieval Using Lemur

We extracted the claims sections from all English patents in the corpus after removing the XML markup from the texts. Since a patent may consist of multiple XML documents, which correspond to the different stages of the patent realization process, one patent can contain more than one claims section. In the index file, we concatenated the claims sections pertaining to one patent ID into one document. We saved all patent claims in the Lemur index format with the patent IDs as DOCIDs. One entry in the index looks like this:

```
<DOC><DOCNO>EP-0148743</DOCNO>
<TEXT> A thermoplastic resin composition comprising a melt mixed
product of (A) 70% to 98% by weight of at least one thermoplastic
resin selected from the group consi sting of polyamides, polyacetals,
polyesters, and polycarbonates and (B) 30% to 2% by weight of
a modified ultra-high molecular weight polyolefin powder having
an average powder particle size of 1 to 80 m and having a particle
size distribution such that substantially all of the powder particles
pass through a sieve having a sieve mesh of 0.147 mm and at least
20% by weight of the total powder particles pass through a sieve
having a sieve mesh of 0.041 mm, said polymer being modified
by graft copolymerizing unsaturated carboxylic acid derivative
units having at least one polar group selected from the group
consisting of acid groups, acid anhydride group,and ester groups
and derived from an unsaturated carboxylic acid or the acid an-
hydride, the salt, or the ester thereof to ultra-high molecular
weight polyolefin having an intrinsic viscosity [#] of 10 dl/g
or more, measured in decalin at 135C.
</TEXT>
</DOC>
```

The claims sections were then indexed using Lemur's BuildIndex function with the Indri IndexType and a stop word list for general English. The batch retrieval and ranking was then performed using the TF-IDF ranking model as it has been included in Lemur. We did not compare the different ranking models provided

by Lemur to each other since the goal of our research is not to find the optimal ranking model[6] but to explore the possibilities and difficulties of any BOW approach.

3 Results

We performed runs for the Main and English Task with the methodology described above. Since we used the same data for both runs, we obtained the same results. These results are in Table 1. The first row shows the results that are obtained if all relevant assignments are taken into consideration; the second row contains the results for the highly-relevant citations only [8].

Table 1. Results for the clefip-run 'ClaimsBOW' on the small topic set using English claims sections for both the Main Task and the English Task

	P	P5	P10	P100	R	R5	R10	R100	MAP	nDCG
All	0.0129	0.0668	0.0494	0.0129	0.2201	0.0566	0.0815	0.2201	0.0540	0.4567
Highly-relevant	0.0080	0.0428	0.0314	0.0080	0.2479	0.0777	0.1074	0.2479	0.0646	0.4567

4 Discussion

Although the results we obtained with our ClaimsBOW approach may seem poor on first sight, they are not bad compared to the results obtained by other participants. In terms of nDCG, our run performs well (ranked 6th of 70 runs); in terms of MAP our results are moderate (ranked around 35th of 70 runs). The low performance achieved by almost all runs (except for one submitted by Humboldt University) shows that the task at hand is a difficult one.

There are a number of reasons for these low scores: First of all, some of the documents were 'unfindable': 17% of the patent documents in the collection contained so little information, e.g. only the title which is poorly informative for patent retrieval [9], that they could not be retrieved. Secondly, the relevance assessments were based on search reports and the citations in the original patent only. This means that they were incomplete [1].

Finally, in order to perform retrieval on the patent level, instead of the document level, some of the participating groups created 'virtual patents': for each field in the patent the most recent information was selected from one of the documents with that patentID. These fields were glued together to form one whole 'virtual' patent. It is, however, not necessarily true that the most recent fields are the most informative [9]. This selection may have resulted in a loss of information. However, even without these impediments, it is clear that patent retrieval is a difficult task for standard retrieval methods.

[6] Such experiments where conducted by the BiTeM group who also participated in this track [7].

The discussion at the CLEF-IP meeting showed that merely text-based retrieval is not enough for patent retrieval. Those groups that made use of the metadata in the patent documents (e.g. classification information) scored remarkably better than those relying on standard text-based methods.

5 Conclusion

The CLEF-IP track was very valuable to us as we now have a baseline that is based on standard bag-of-words text retrieval techniques. In future work we are going to focus on improving the ranking of the result list that we produced in the CLEF-IP experiment. We plan to apply an additional reranking step to the result set using syntactic information in the form of dependency triplets [10].

References

1. Roda, G., Tait, J., Piroi, F., Zenz, V.: CLEF-IP 2009: retrieval experiments in the Intellectual Property domain. In: CLEF working notes 2009 (2009)
2. Piroi, F., Roda, G., Zenz, V.: CLEF-IP 2009 Track Guidelines. Technical report, Information Retrieval Facility (2009)
3. Graf, E., Azzopardi, L.: A methodology for building a test collection for prior art search. In: Proceedings of the 2nd International Workshop on Evaluating Information Access (EVIA), pp. 60–71 (2008)
4. Shinmori, A., Okumura, M., Marukawa, Y., Iwayama, M.: Patent claim processing for readability: structure analysis and term explanation. In: Proceedings of the ACL-2003 workshop on Patent corpus processing, pp. 56–65. Association for Computational Linguistics, Morristown (2003)
5. Iwayama, M., Fujii, A., Kando, N., Marukawa, Y.: Evaluating patent retrieval in the third NTCIR workshop. Information Processing Management 42(1), 207–221 (2006)
6. Graf, E., Azzopardi, L., Van Rijsbergen, K.: Automatically Generating Queries for Prior Art Search. In: CLEF working notes 2009 (2009)
7. Gobeill, J., Theodoro, D., Ruch, P.: Exploring a wide Range of simple Pre and Post Processing Strategies for Patent Searching in CLEF IP 2009. In: CLEF working notes 2009 (2009)
8. Piroi, F., Roda, G., Zenz, V.: CLEF-IP 2009 Evaluation Summary. Technical report, Information Retrieval Facility (2009)
9. Tseng, Y., Wu, Y.: A study of search tactics for patentability search: a case study on patent engineers. In: Proceeding of the 1st ACM workshop on Patent information retrieval, pp. 33–36 (2008)
10. D'hondt, E., Verberne, S., Oostdijk, N., Boves, L.: Re-ranking based on Syntactic Dependencies in Prior-Art Retrieval. In: Proceedings of the Dutch-Belgium Information Retrieval Workshop (to appear, 2010)

UniGE Experiments on Prior Art Search in the Field of Patents

Jacques Guyot[1], Gilles Falquet[1], and Karim Benzineb[2]

[1] Computer Science Center, University of Geneva, Route de Drize 7, 1227 Carouge, Switzerland
[2] Simple Shift, Ruelle du Petit-Gris 1, 1228 Plan-les-Ouates, Switzerland
{jacques.guyot,gilles.falquet}@unige.ch, karim@simple-shift.com

Abstract. In this experiment led at the University of Geneva (UniGE), we evaluated several similarity measures as well as the relevance of using automated classification to filter out search results. The patent field is particularly well suited to classification-based filtering because each patent is already classified. Our results show that such a filtering approach does not improve searching performances, but it does not have a negative impact on recall either. This last observation allows considering classification as a possible tool to reduce the search space without reducing the quality of search results.

1 Introduction

Our task was to automatically identify all the quotes included in a given patent. A quote was defined as a reference to another patent. All the quotes originated from a target corpus containing about 2 million documents; the patent corpus itself included one million patents. For each patent, several documents could be available, reflecting the various filing stages. The total size of the corpus was 75 Gb. In average, each patent included less than 10 quotes. In the evaluation phase, we had to identify the quotes in 10'000 patents (hereafter called "topics"). In the test phase, a patent collection containing pre-identified quotes allowed to fine-tune the various systems. A complete description of the task is available in [1].

Approach. We used our VLI indexer, which is described in [2]. This indexer allows performing a similarity search with the standard cosine methodology. The results can be weighted out in three possible ways:

- TF-IDF evaluates the weight according to a term frequency in the document related to its inverse frequency with regards to the number of retrieved documents [3];
- FAST is based on the hypothesis that TF is always equal to 1;
- OKAPI is a weighted version of TF-IDF [4].

The results were then filtered with an automated classifier, retaining only those results whose classes match the classes of the query. For the classification job, we used a classifier based on a neural network algorithm of the Winnow type [5]. This classifier is well suited to the hierarchical structure of patent classification.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 502–507, 2010.

2 Implementation

The first experiment aimed at setting up a baseline to validate the following experiments. For that purpose we used the entire corpus (75 Gb) with the FAST weighting and without any filtering process. The MAP score of this experiment was 4.67 (all results in this article are calculated with regards to the standard test collection).

In the second experiment we introduced classification-based filtering. On the basis of the target corpus, we built a catalogue of the categories affected to each document. For the training phase, it was difficult to use the target corpus since a number of documents in that corpus appeared several times. This was probably due to the fact that classification could vary throughout the successive filing stages. Thus we decided to train the system on another patent corpus whose classification quality and performance as a training basis had been previously checked out. The classification in that corpus was based on WIPO's International Patent Classification (IPC), while the classification of the target corpus was based on the EPO's European Classification system (ECLA). Fortunately, both classifications are compatible down to the sub-class level. Thus we could choose between three possible classification levels: *Section* (8 categories), *Class* (~120 categories), and *Sub-class* (~620 categories.)

The filtering process was performed in the following way:

- The classifier was required to predict the *M* most probable categories for a given topic[1]
- Based on the topic, the most similar documents were searched for in the target corpus
- The retrieved documents were kept if and only if the intersection of the document categories and the topic categories was not empty.

When starting over from the entire target corpus, we got a MAP score of 5.26 if the filtering process predicted 3 categories, and of 5.34 on 6 categories (at the Sub-class level). Thus such a filtering approach did improve the performance, but only slightly. We also performed the same tests at Class level, with no result improvement. Therefore we kept filtering at Sub-class level in all our following experiments.

Since our task was focused on an English corpus, we removed the French or German parts of the training corpus. Additionally, we selected a number of specific fields to be indexed:

- The English "Title" of the invention;
- The English "Abstract" of the invention;
- The English "Claims" of the invention.

We thus produced a focused Target Corpus ("TAC"). We also filtered out the topics. A new experiment with no additional filtering produced a MAP score of 4.43, i.e. slightly degraded with regards to the MAP score on the full corpus. A close look at the search results showed that some documents only contained a title. It could happen that the title words were included in the topic words, in which case the document

[1] Performance evaluation on the WIPO corpus showed that at least one of the correct categories belongs to the 3 most probable categories with probability 0.8.

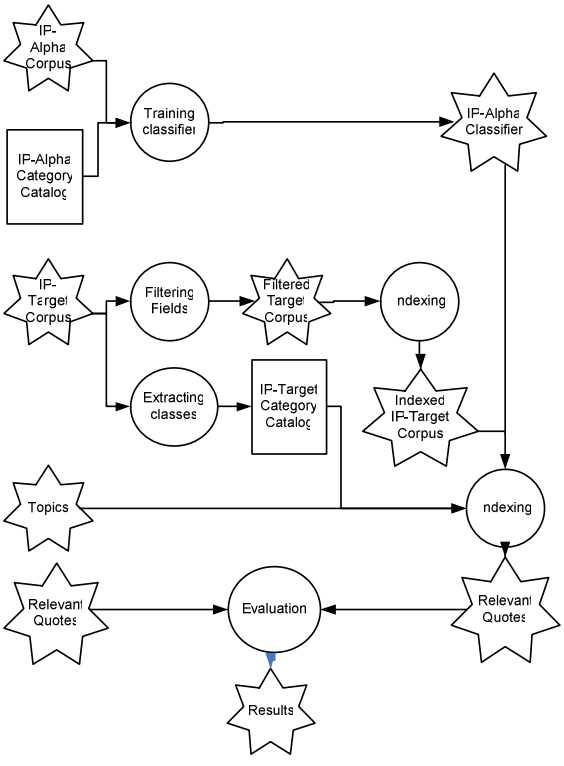


Fig. 1. Search Process Overview

would be rated as highly similar. In order to get rid of the documents that were deemed too small, we added a filter on the length L (in words) of the retrieved documents. By successive tests it came out that $L = 25$ seemed to maximize the performance, with a MAP score of 6.13.

When cumulating the filterings on length and classification, we got a MAP score of 6.32 with a prediction of 6 categories. By exploring other combinations, we noted that $M = 4$ provided the best results with a MAP of 6.33.

On the basis of this last result, we added stemming to the indexing process. Stemming allows merging the various morphological aspects of a given word into a single token (e.g. merging the singular and plural forms). We got a MAP score of 5.33, so stemming seems to weaken the performance. More precisely, stemming is efficient on small corpuses but it generally introduces confusion in larger collections.

In the next experiment, we modified the similarity weighting method by using TF-IDF. This approach is more costly in terms of indexing and calculation time because it needs to assess the number of word occurrences per document. Our implementation of TF-IDF included two parameters, *MinOcc* and *Max%*, to eliminate the words which were either too frequent or too rare. With *MinOcc* = 2 and *Max%* = 10 we got a MAP score of 9.21; thus such a weighting approach did improve the search precision.

Finally we added to the target corpus the "Applicant" field and all the fields available in other languages, in order to stop filtering on the topic and to use the full data (since some English topics had been translated). The resulting corpus ("ATAC") included:

- The "Applicant" and "Inventors" fields
- The invention "Title" in the three languages (if available)
- The invention "Abstract" in the three languages (if available)
- The invention "Claims" in the three languages (if available)

This experiment produced a MAP score of 10.27 (with *MinOcc* = 2 and *Max%* = 5, *M* = 4, *L* = 25).

We also checked out that the results were poorer if the topic was reduced to the ATAC fields. Besides, we checked out that our classifier did not introduce any bias with regards to the classification of the target corpus; to do so we reclassified the 2 million patents of the target corpus. The MAP score of this experiment was 9.74. However, reclassifying the corpus introduced some inaccuracies which slightly disturbed the filtering process. We also tried out the OKAPI weighting methodology on similarity calculation, but this did not produce any positive result.

In our best experiment, about 50% of the documents to be found appeared in our result set (of 1'000 documents). The problem was that these documents were way down in our result list. The similarity method we used did not take into account the word order, since a document was considered as a "bag of words". Thus we tried out two other methods of similarity calculation which were based on word sequences: the Kolmogorov distance [6] and the Lesk distance [7]. We used those distance calculations to re-order the result vector produced with the methodology described above. The results were poorer than when using the original order.

The runs that were sent out to the CLEF-IP team for the evaluation phase were the following:

- **Clefip-unige RUN1 (MAP 10.52%)** Similarity: **TF-IDF**, length filtering: **yes**, category filtering: **yes**
- **Clepip-unige RUN2 (MAP 10.58%)** Similarity: **TF-IDF**, length filtering: **yes**, category filtering: **no**
- **Clepip-unige RUN3 (MAP 10.52%)** Similarity: **TF-IDF**, length filtering: **no**, category filtering: **yes**
- **Clepip-unige RUN4 (MAP 7.49%)** Similarity: **OKAPI**, length filtering: **yes**, category filtering: **yes**
- **Clepip-unige RUN5 (MAP 7.61%)** Similarity: **FAST**, length filtering: **yes**, category filtering: **yes**

The evaluation run results were in line with what had been measured during the test phase.

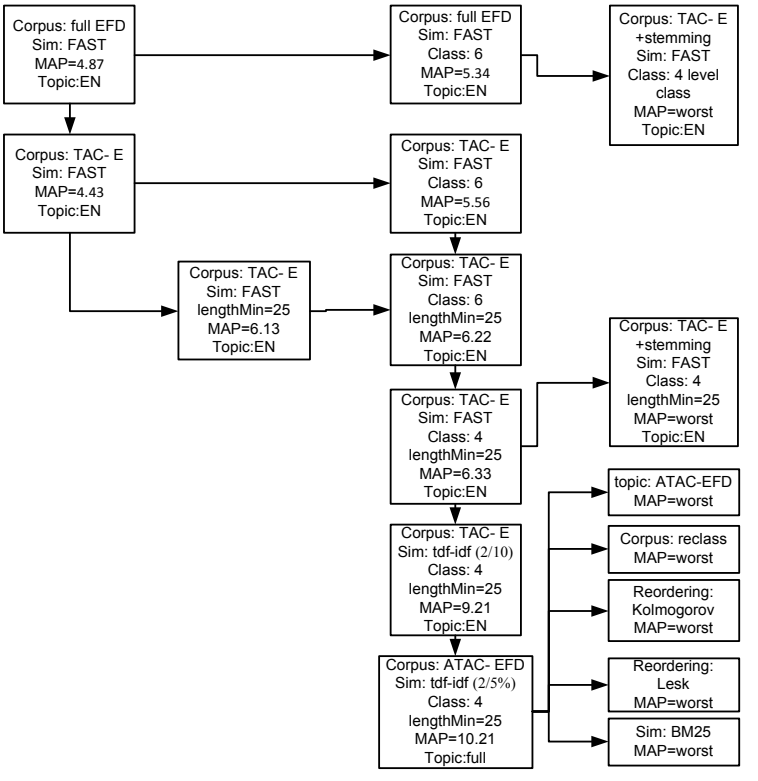


Fig. 2. Parameters and Results of Each Experiment

3 Findings and Discussion

We were rather disappointed to see that category filtering did not eliminate much noise. Such a filtering approach is only efficient when the request is short. In this case, the request being a complete patent, the classification-based filtering did not help because the similarity calculated with the cosine implicitly acted as a *k*NN (*k* Nearest Neighbours) algorithm, which is an alternative to automated classification. Filtering on length during the test runs did improve the performance to some extent, but on the final runs it seemed to have no effect whatsoever. As for the runs which relied on other weighting methodologies, their performance was poorer than the standard approach.

The positive aspects of this experiment are the following:

- The implementation of our similarity search algorithms was efficient: The processing time of a topic was about one second on a standard PC.
- Filtering on the basis of automated classification does not directly improve performance. However, it can be used to indirectly improve performance if the index

is broken down in clusters for each category. In such a case, the search time is divided by the total number of categories in the classification, and then multiplied by the number of categories in the filter. In our experiment we had 600 categories in the classification and 4 categories in the filter, so we had a ratio of 150 (in the hypothesis of an even distribution across all categories). It should be emphasized that breaking down the index is only useful on a corpus which is much larger than the one used in our experiments.

– Yet classification-based filtering does improve performance when the method used to calculate similarity is not optimal (which is the case with the FAST approach) because it limits the number of accepted result categories and thus provides for a better relevance of the search results.

References

1. Piroi, F., Roda, G., Zenz, V.: CLEF-IP 2009 Track Guidelines (2009), http://www.irf.com
2. Guyot, J., Falquet, G., Benzineb, K.: Construire un moteur d'indexation. Technique et science informatique (TSI), Hermes, Paris (2006)
3. Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill, New York (1983)
4. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Proceedings of the Third Text REtrieval Conference (TREC 1994), Gaithersburg, USA (1994)
5. Fall, C.J., Benzineb, K., Guyot, J., Törcsvéri, A., Fiévet, P.: Computer-Assisted Categorization of Patent Documents in the International Patent Classification. In: Proceedings of the International Chemical Information Conference (ICIC 2003), Nîmes, France (2003)
6. Ming, L., Vitányi, P.: An Introduction to Kolmogorov Complexity and Its Applications. Springer, Heidelberg (1997)
7. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In: Proceedings of SIGDOC 1986 (1986)

LogCLEF 2009: The CLEF 2009 Multilingual Logfile Analysis Track Overview

Thomas Mandl[1], Maristella Agosti[2], Giorgio Maria Di Nunzio[2], Alexander Yeh[3], Inderjeet Mani[3], Christine Doran[3], and Julia Maria Schulz[1]

[1] Information Science, University of Hildesheim, Germany
{mandl,schulzju}@uni-hildesheim.de
[2] Department of Information Engineering, University of Padua, Italy
{agosti,dinunzio}@dei.unipd.it
[3] MITRE Corporation, Bedford, Massachusetts, USA
{asy,imani,cdoran}@mitre.org

Abstract. Log data constitute a relevant aspect in the evaluation process of the quality of a search engine and the quality of a multilingual search service; log data can be used to study the usage of a search engine, and to better adapt it to the objectives the users were expecting to reach. The interest in multilingual log analysis was promoted by the Cross Language Evaluation Forum (CLEF) for the first time with a track named LogCLEF. LogCLEF is an evaluation initiative for the analysis of queries and other logged activities as expression of user behavior. The goal is the analysis and classification of queries in order to understand search behavior especially in multilingual contexts and ultimately to improve search systems. Two tasks were defined: Log Analysis and Geographic Query Identification (LAGI) which aimed at the identification of queries for geographic content and Log Analysis for Digital Societies (LADS) which was based on analyzing the user behavior of the search logs the service of The European Library. Five groups using a variety of approaches submitted experiments. The data for the track, the evaluation methodology and results are presented and discussed.

1 Introduction

Logging is a concept commonly used in computer science; in fact, log data are collected by an operating system to make a permanent record of events during the usage of the operating system itself. This is done to better support its operations, and in particular its recovery procedures. Log data are also collected by many applications systems that manage permanent data, among the more relevant ones there are the database management systems (DBMS) that support different types of collection of log data, one of these types is the Write Ahead Log (WAL) that is used for the management and the recovery of transactions. Due to the experience gained in the management of operating systems and the many other application systems that manage permanent data, log procedures are commonly put in place to collect and store data on the usage of application systems by its users. Initially, these data were

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 508–517, 2010.

mainly used to manage recovery procedures of an application system, but over time it became apparent that they could also be used to study the usage of the application by its users, and to better adapt the system to the objectives the users were expecting to reach.

Like with an operating system and any other software application, log data can be collected during the use of a search engine to monitor its functioning and usage by final and specialized users, which means recording log data to study its use and to consolidate it in a tool that meets end-user search requirements. This means that log data constitute a relevant aspect in the evaluation process of the quality of a search engine and the quality of a multilingual search services; log data can be used to study the usage of a search engine, and to better adapt it to the objectives the users were expecting to reach.

The interest in multilingual log analysis was promoted by the Cross Language Evaluation Forum (CLEF)[1] for the first time with a track named LogCLEF[2] which is an evaluation initiative for the analysis of queries and other logged activities as expression of user behavior. The main goal of LogCLEF is the analysis and classification of queries in order to understand search behavior in multilingual contexts and ultimately to improve search systems. Another important long-term aim is to stimulate research on user behavior in multilingual environments and promote standard evaluation collections of log data.

The datasets used in 2009 for the analysis of search logs were derived from the use of the Tumba! Search engine and The European Library (TEL) Web site[3]. Two tasks were defined: Log Analysis and Geographic Query Identification (LAGI) and Log Analysis for Digital Societies (LADS). LAGI required the identification of geographical queries within logs from the Tumba! Search engine and The European Library multilingual information system. LADS intended to analyze the user behavior in the multilingual information system of The European Library. Five groups using a variety of approaches submitted experiments.

The data for the track, the evaluation methodology and some results are presented in this overview paper together with a description of the two sub tasks of LogCLEF 2009.

2 Log Analysis and Geographic Query Identification (LAGI)

The identification of geographic queries within a query stream and the recognition of the geographic component are key problems for geographic information retrieval (GIR). Geographic queries require specific treatment and often a geographically oriented output (e.g. a map). The task would be to (1) classify geographic queries and (2) identify their geographic content. The task design and evaluation measures would be similar to the ones used in the track in 2007 [10, 11].

[1] http://www.clef-campaign.org/
[2] http://www.uni-hildesheim.de/logclef/
[3] http://www.theeuropeanlibrary.org/

LAGI 2009 is a task to identify geographic elements in query search logs. Search logs were obtained from the following two search systems:

1. Tumba!, a Portuguese web search engine[4] (350.000 queries)
2. The on-line catalogue of The European Library (TEL), where an English subset of the TEL queries was used[5] (1.8 million records).

The Tumba! log files were manually reviewed and anonymized with a custom-made program following the rules presented by Korolova and colleagues [13]. All references to domain searches and emails were replaced by 'ZZZ' in order to preserve anonymity. Examples for entries in the log file are shown in Tables 1a to 1d.

Table 1a. Original queries from the TEL log file

```
875336 & 5431 & ("central europe")
828587 & 12840 & ("sicilia")
902980 & 482 &  (creator all "casanova")
196270 & 5365 & ("casanova")
906474 & 15432 & casanova
528968 & 190 &  ("iceland*")
470448 & 8435 & ("iceland")
712725 & 5409 & ("cavan county ireland 1870")
671397 & 14093 & ("university")
```

Table 1b. Annotated queries from the TEL log file

```
875336 & 5431 & ("<place>central europe</place>")
828587 & 12840 & ("<place>sicilia</place>")
902980 & 482 &  (creator all "casanova")
196270 & 5365 & ("casanova")
906474 & 15432 & casanova
528968 & 190 &  ("<place>iceland</place>*")
470448 & 8435 & ("<place>iceland</place>")
712725 & 5409 & ("<place>cavan county ireland</place> 1870")
671397 & 14093 & ("university")
```

Table 1c. Original queries from the Tumba! log file

```
4333825 @ 4777 @ "administração escolar"
4933229 @ 7888 @ "escola+hip+hop"
39283 @ 62180 @ chaves
2290106 @ 19398 @ CHAVES
1420489 @ 20564 @ Chaves
6971716 @ 106342 @ jornais de leiria
8403308 @ 83318 @ escolas de marinheiro
```

[4] http://www.tumba.pt/
[5] http://www.theeuropeanlibrary.org/

Table 1d. Annotated queries from the Tumba! log file

```
4333825 @ 4777 @ "administração escolar"
4933229 @ 7888 @ "escola+hip+hop"
39283 @ 62180 @ <place>chaves</place>
1420489 @ 20564 @ <place>Chaves</place>
6971716 @ 106342 @ jornais de <place>leiria</place>
8403308 @ 83318 @ escolas de marinheiro
```

The geographic elements to be marked are either those found in a gazetteer or places related to those found in the gazetteer. So if a city is listed in a gazetteer, then related places include hospitals, schools, etc. associated with that city. The gazetteer used is a static version of Wikipedia (Portuguese version for Tumba!, English version for the TEL subset), due to its coverage and availability to participants. A static version is used because the live Wikipedia (http://pt.wikipedia.org, http://en.wikipedia.org) is constantly changing, and so altering what places are listed, etc.

Many query terms have both geographic and non-geographic senses. Examples include 'casanova' and 'ireland' in English and 'chaves' in Portuguese. Queries have inconsistent capitalization and are often short. So it may not be clear which sense to use for such terms. Wikipedia is used to disambiguate such terms by preferring the first sense returned by a Wikipedia look-up. In our examples, a look-up[6] of 'casanova' initially returns an article on a person, so it is deemed a non-place in ambiguous cases. A look-up of 'Ireland' initially returns an article on an island, so it is deemed a place in ambiguous cases. Sometimes, the initial article returned does not have a clear preference. For example, a look-up of 'chaves' returns a disambiguation page/article which lists both place (including a city) and non-place (including a television show) senses. In such situations, the term is deemed a place in ambiguous cases for the purposes of this evaluation. This method of disambiguation is used when a query has no indicated preference for which sense to use. But if the query indicates a preference for a sense, then that sense is what is used. An example is the query 'casanova commune'. A search for 'casanova commune' in the English Wikipedia does not return an article. Rather it returns a 'search' page (instead of being an article for some term, the page gives a ranked list of articles that contain parts of the candidate place term somewhere in the articles' text), which is ignored in this evaluation. For 'casanova', the English Wikipedia returns an article on a person named Casanova, so that is the default predominant sense. But that article has a link to a disambiguation page, and the disambiguation page has a link to the place 'Casanova, Haute-Corse', which is a commune. This query indicates that this sense of 'casanova' is the preferred one for the query, so this overrides the default preferred sense based on the initial page returned by the Wikipedia.

There are still complications in look-ups. For one thing, it turns out that many queries have misspelled words, and judgments need to be made about these. Also, many terms exist in queries that turn out not to have a Wikipedia article about them. In addition, Wikipedia will sometimes prefer an unusual sense of a word over a more

[6] To look-up a term (not search for a term), type the term into the Wikipedia 'search' (English) or 'busca' (Portuguese) box and then click 'Go' (English) or 'Ir' (Portuguese).

usual sense. Two examples are “de” in Portuguese and “Parisian” in English. “de” is a common preposition meaning something like “of” . For example, “jornais de leiria” loosely means “periodicals of leiria”. A look-up of “de” returns a disambiguation page that mentions “de” possibly standing for Delaware or Deutschland (Germany), but does not mention “de” being a preposition. Various common meanings for “Parisian” include a person from Paris or something in or associated with Paris. But a look-up of “Parisian” first returns an article about a chain of department stores in the US with the name “Parisian”. We dealt with these complications by adding to the task guidelines and removing queries that could not be handled by the guidelines.

Beyond look-up complications, there were also complications with Wikipedia software. It turns out that installing a static version of Wikipedia is hard. One group made an unsuccessful attempt and another group could install it well enough to support the evaluation, but there were still complications. The successful installation was served over the Internet for others. Besides being hard to install, a Wikipedia is somewhat slow as it takes quite a bit of computational resources to run and Internet congestion can considerably add to the response time.

These complications of Wikipedia combined with difficulties in obtaining the search logs until late in the CLEF campaign delayed the dataset annotation and constrained its size: there was only enough annotated data to produce a small test set (and no training set). The TEL test set has 108 queries with 21 places annotated. The Tumba! test set has 146 queries and 30 to 35 places annotated[7].

A total of two runs were submitted, both by the same group at the “Alexandru Ioan Cuza” University in Romania [8]. The two runs used different resources (1. GATE, 2. Wikipedia) for finding places. Overall, precision in finding places turned out to more of challenge than recall. The recall scores ranged from 33% to 76%, while the precision scores were 26% or less.

Table 2. Results of the LAGI sub task for the test set

Resource	TEL	Tumba! version A	Tumba! version B
Gate	R 33%, P 24%	R 51%, P 26%	R 50%, P 22%
Wikipedia	R 76%, P 16%	R 37%, P 9%	R 40%, P 8%

With both TEL and Tumba!, the Wikipedia resources produced much lower precision than the GATE resources. Using the Wikipedia resources often resulted in an entire query being annotated as a place. For Tumba!, the Wikipedia resources also produced worse recall, but for TEL, Wikipedia resources produced better recall.

In summary, this is a first run of the LAGI task for finding places in search queries. We obtained the use of two search query sets, Tumba! and TEL, and used Wikipedia as a gazetteer and for disambiguating between place and non-place senses of a term. Delays in obtaining the data sets and complications with using Wikipedia resulted in a delay in producing a dataset and also only a small test set being produced (no training set).

[7] The Tumba! test set had a number of queries that could be annotated in two different ways. We made two versions where version A was annotated in one way (for 35 places) and version B the other way (for 30 places).

3 Log Analysis for Digital Societies (LADS)

The Log Analysis for Digital Society (LADS) task deals with logs from The European Library (TEL) and intends to analyze user behavior with a focus on multilingual search. TEL is a free service that offers access to the resources of 48 national libraries of Europe in 35 languages, it aims to provide a vast virtual collection of material from all disciplines and offers interested visitors simple access to European cultural heritage. Resources can be both digital (e.g. books, posters, maps, sound recordings, videos) and bibliographical and the quality and reliability of the documents are guaranteed by the 48 collaborating national libraries of Europe[8].

The quality of the services and documents TEL supplies are very important for all the different categories of users of a digital library system; for this reason, log data constitute a relevant aspect in the evaluation process of the quality of a digital library system and of the quality of interoperability of digital library services [1].

The access to the services that access the TEL digital library is supplied through a Web browser, and not through a specifically designed interface. This means that the analysis of user interaction of such a digital library system requires the forecasting of ways that support the reconstruction of sessions in a setting, like the Web, where sessions are not naturally identified and kept [2].

3.1 Goals

Potential targets for experiments are query reformulation, multilingual search behavior and community identification. This task was open to diverse approaches, in particular data mining techniques in order to extract knowledge from the data and find interesting user patterns.

Suggested sub-tasks for the analysis of the log data were:

1. user session reconstruction; this step was considered as a prerequisite to the following ones [3];
2. user interaction with the portal at query time; e.g. how users interact with the search interface, what kind of search they perform (simple or advanced), and how many users feel satisfied/unsatisfied with the first search and how many of them reformulate queries, browse results, leave the portal to follow the search in a national library;
3. multilinguality and query reformulation; e.g. what are the collections that are selected the most by users, how the language (country/portal interface) of the user is correlated to the collections selected during the search, how the users reformulate the query in one language or in a different language;
4. user context and user profile; e.g. how the study of the actions in the log can identify user profiles, how the implicit feedback information recorded in the logs can be exploited to create the context in which the user operates and how this context evolves.

[8] http://www.theeuropeanlibrary.org/portal/organisation/about_us/aboutus_en.html

Participants were required to:

- process the complete logs;
- make publicly available any resources created based on these logs (e.g. annotations of a small subsets);
- find out interesting issues about the user behavior as exhibited in the logs;
- submit results in a structured file.

3.2 Data

The data used for the LADS task are search logs of The European Library portal; those logs are usually named "action logs" in the context of TEL activities. In order to better understand the nature of the action logs that have been distributed to the participants, the following example of possible usage of the portal is given: in TEL portal's home page, a user can initiate a simple keyword search with a default predefined collection list presenting catalogues from national libraries. From the same page, a user may perform an advanced search with Boolean operators and/or limit search to specific fields like author, language, and ISBN. It is also possible to change the searched collection by checking the theme categories below the search box. After search button is clicked the result page appears, where results are classified by collections and the results of the top collection in the list are presented with brief descriptions. Then, a user may choose to see result lists of other collections or move to the next page of records of current collection's results. While viewing a result list page a user may also click on a specific record to see detailed information about the specific record. Additional services may be available according to the record selected.

All these type of actions and choices are logged and stored by TEL in a relational table, where each record represents a user action. The most significant columns of the table are:

- A numeric id, for identifying registered users or "guest" otherwise;
- User's IP address;
- An automatically generated alphanumeric, identifying sequential actions of the same user (sessions) ;
- Query contents;
- Name of the action that a user performed;
- The corresponding collection's alphanumeric id;
- Date and time of the action's occurrence.

Action logs distributed to the participants of the task cover the period from 1st January 2007 until 30th June 2008. The log file contains user activities and queries entered at the search site of TEL. Examples for entries in the log file are shown in Table 3.

Table 3. Examples from the TELlog (date has been deleted for readability)

```
id;userid;userip;sesid;lang;query;action;colid;nrrecords;recordposition;sboxid;objurl;date
892989;guest;62.121.xxx.xxx;btprfui7keanue1u0nanhte5j0;en;("plastics mould");view_brief;a0037;31;;;
893209;guest;213.149.xxx.xxx;o270cev7upbblmqja30rdeo3p4;en;("penser leurope");search_sim;;0;-;;;
893261;guest;194.171.xxx.xxx;null;en;(“magna carta”);search_url;;0;-;;;
893487;guest;81.179.xxx.xxx;9rrrtrdp2kqrtd706pha470486;en;("spengemann");view_brief;a0067;1;-;;;
893488;guest;81.179.xxx.xxx;9rrrtrdp2kqrtd706pha470486;en;("spengemann");view_brief;a0000;0;-;;;
893533;guest;85.192.xxx.xxx;ckujekqff2et6r9p27h8r89le6;fr;("egypt france britain");search_sim;;0;-;;;
```

3.3 Participants and Experiments

As shown in Table 4, a total of 4 groups submitted results for the LADS task. The results of the participating groups are reported in the following section.

Table 4. LogCLEF 2009 participants

Participant	Institution	Country
Sunderland	University of Sunderland	UK
TCD-DCU	ï ïïï ïïï ïï ï ïïï ï ï ïï ï ï ïïï ïï	ïïïï ïï ï ï ï
Info Science	University of Hildesheim	Germany
CELI s.r.l	CELI Research, Torino	Italy

3.4 Results of the LADS Task

The CELI research institute tried to identify translations of search queries [4]. The result is a list of pairs of queries in two languages. This is an important step in observing multilingual user behavior. Combined with session information, it is possible the check whether users translate their query within a session. The analysis showed the true multilingual nature of the data.

The group from the University of Sunderland argues that users rarely switch the query language during their sessions. They also found out that queries are typically submitted in the language of the interface which the user selects [12]. An exception is, or course, English which is the default language of the interface and as in any user interface, the default is not always modified.

A thorough analysis of query reformulation, query length and activity sequence was carried out by the Trinity College, Dublin [6]. The group showed that amny query modification operations concern the addition or the removal of stopwords. These actions only have an effect for the language collection in which the word is a stop word. The ultimate goal is the understanding of the behavior of users from different linguistic or cultural backgrounds. The application of activity sequences for the identification of communities is also explored. The analysis revealed the most frequent operations as well as problems with the user interface of TEL.

The University of Hildesheim analyzed sequences of interactions within the log file. These were visualized in an interactive user interface which allows the exploration of the sequences [9]. In combination with a heuristic success definition, this system lets one identify typical successful activity sequences. This analysis can be done for users from one top level domain. A few differences for users from different

countries were observed but more analysis is necessary to reveal if these are real differences in behavior. In addition, issues with the logging facility were identified.

The design of future tasks is encouraged by a position paper from the University of Amsterdam. The authors argue that the limited knowledge about the user which is inherent in log files needs to be tackled in order to gain more context information. They argue for the semantic enrichment of the queries by linking them to digital objects [7].

4 Conclusions and Future Work

Studies on log files are essential for personalization purposes, since they implicitly capture user intentions and preferences in a particular instant of time. There is an emerging research activity about log analysis which tackles cross-lingual issues: extending the notion of query suggestion to cross-lingual query suggestion studying search query logs; leveraging click-through data to extract query translation pairs.

LogCLEF has provided an evaluation resource with log files of user activities in multilingual search environments: the Tumba! Search engine and The European Library (TEL) Web site. With these two different datasets, one related with searches in Web sites of interest to the Portuguese community and the other with searches for library catalogues in many European libraries, it was possible to define two sub-tasks: Log Analysis and Geographic Query Identification (LAGI) and Log Analysis for Digital Societies (LADS).

For LADS, a total of 4 groups submitted a very diverse set of results: identifying a list of pairs of queries in two languages combined with session information; correlation between language of the interface and language of the query; activities at query time to study different user backgrounds.

Given the success of LogCLEF 2009, considering the fact that it was a pilot task, and the good feedback from people who were at the workshop and also from those who could not participate, a second LogCLEF workshop will be organized as a research lab in the CLEF2010 conference. New logs will be offered, in particular:
- action logs from January 2007 to June 2008 (same logs of LogCLEF 2009);
- http logs from January 2007 to June 2008 (new, more than 50 millions records);
- action logs from January 2009 to December 2009 (new, more than 700,000 records).

Acknowledgments

The organization of LogCLEF was mainly volunteer work. At MITRE, work on this task was funded by the MITRE Innovation Program (public release approval case# 09-3930 - Paperwork update). We want to thank The European Library (TEL) and the Tumba! search engine for providing the log files. Many thanks especially to Nuno Cardoso from the XLDB Research Team at the University of Lisbon, who coordinated the anonymisation and the manual reviewing of the original logfile. At the University of Padua, work on this task was partially supported by the TELplus Targeted Project for digital libraries[9], as part of the eContent*plus* Program of the European

[9] http://www.theeuropeanlibrary.org/telplus/

Commission (Contract ECP-2006-DILI-510003) and by the TrebleCLEF Coordination Action[10], as part of the 7th Framework Program of the European Commission, Theme ICT-1-4-1 Digital libraries and technology-enhanced learning (Grant agreement: 215231).

References

1. Agosti, M.: Log Data in Digital Libraries. In: Agosti, M., Esposito, F., Thanos, C. (eds.) Post-proceedings of the Fourth Italian Research Conference on Digital Library Systems (IRCDL 2008), July 2008, pp. 115–121. An Association for Digital Libraries, Padova (2008)
2. Agosti, M., Di Nunzio, G.M.: Gathering and Mining Information from Web Log Files. In: Thanos, C., Borri, F., Candela, L. (eds.) Digital Libraries: Research and Development. LNCS, vol. 4877, pp. 104–113. Springer, Heidelberg (2007)
3. Agosti, M., Di Nunzio, G.M.: Web Log Mining: A Study of User Sessions. 10th DELOS Thematic Workshop on Personalized Access, Profile Management, and Context Awareness in Digital Libraries (PersDL, Corfu, Greece, pp. 70-74 (2007)
4. Bosca, A., Dini, L.: CACAO project at the LogCLEF Track. In: this volume
5. Di Nunzio, G.M.: LogCLEF, 2009/03/02 v 1.0 Description of the The European Library (TEL) Search Action Log Files (2009), http://www.uni-hildesheim.de/logclef/Daten/LogCLEF2009_file_description.pdf
6. Ghorab, M.R., Leveling, J., Zhou, D., Jones, G., Wade, V.: TCD-DCU at LogCLEF 2009: An Analysis of Queries, Actions, and Interface Languages. In: this volume
7. Hofmann, K., de Rijke, M., Huurnink, B., Meij, E.: A Semantic Perspective on Query Log Analysis, http://www.clef-campaign.org/2009/working_notes/hofmann_etal-clef2009-querylog.pdf
8. Iftene, A.: UAIC: Participation in LAGI Task. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 362–365. Springer, Heidelberg (2010)
9. Lamm, K., Mandl, T., Kölle, R.: Search Path Visualization and Session Performance Evaluation with Log Files from The European Library (TEL). In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 539–544. Springer, Heidelberg (2010)
10. Li, Z., Wang, C., Xing, X., Ma, W.-Y.: Query Parsing Task for GeoCLEF 2007 Report. In: Cross Language Evaluation Forum, CLEF (2007), Working Notes, http://www.clef-campaign.org/2007/working_notes/LI_OverviewCLEF2007.pdf
11. Mandl, T., Gey, F., Di Nunzio, G.M., Ferro, N., Larson, R., Sanderson, M., Santos, D., Womser-Hacker, C., Xing, X.: GeoCLEF 2007: the CLEF 2007 Cross-Language Geographic Information Retrieval Track Overview. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 745–772. Springer, Heidelberg (2008)
12. Oakes, M., Xu, Y.: LADS at Sunderland. In: this volume
13. Korolova, A., Kenthapadi, K., Mishra, N., Ntoulas, A.: Releasing search queries and clicks privately. In: Proc 18th intl Conference on World Wide Web, Madrid, Spain, April 20 - 24, pp. 171–180. ACM, New York (2009)

[10] http://www.trebleclef.eu/

Identifying Common User Behaviour in Multilingual Search Logs

M. Rami Ghorab[1], Johannes Leveling[2], Dong Zhou[1], Gareth J.F. Jones[2], and Vincent Wade[1]

[1] Centre for Next Generation Localisation
Knowledge and Data Engineering Group
Trinity College Dublin
Dublin 2, Ireland
{ghorabm,dong.zhou,vincent.wade}@scss.tcd.ie
[2] Centre for Next Generation Localisation
School of Computing
Dublin City University
Dublin 9, Ireland
{jleveling,gjones}@computing.dcu.ie

Abstract. The LADS (Log Analysis for Digital Societies) task at CLEF aims at investigating user actions in a multilingual setting. We carried out an analysis of search logs with the objectives of investigating how users from different linguistic or cultural backgrounds behave in search, and how the discovery of patterns in user actions could be used for community identification. The findings confirm that users from a different background behave differently, and that there are identifiable patterns in the user actions. The findings suggest that there is scope for further investigation of how search logs can be exploited to personalise and improve cross-language search as well as improve the TEL search system.

1 Introduction

The Log Analysis for Digital Societies (LADS) task is part of the LogCLEF track at the Cross-Language Evaluation Forum (CLEF) 2009. The LADS dataset contains log entries of user interactions with the TEL[1] portal. The logs were analysed to investigate the following hypotheses: (1) users from different linguistic or cultural backgrounds behave differently in search; (2) there are patterns in user actions which could be useful for stereotypical grouping of users; (3) user queries reflect the mental model or prior knowledge of a user about a search system.

The remainder of this paper is organised as follows: Sect. 2 gives a brief description of the logs, Sect. 3 discusses the log analysis and results, and the paper ends with conclusions and outlook to future work in Sect. 4.

[1] http://www.theeuropeanlibrary.org/

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 518–525, 2010.

2 Brief Description of Logs and Preprocessing Operations

A log entry is created for every user interaction with the TEL portal. Log entries contain the type of action performed, together with attributes such as the interface language, query, and timestamp. The experiments focused on the following attributes: *lang* (interface language selected by the user), *action*, and *query*. The main actions that our study investigated were:

- search_sim: searching via a simple text box.
- search_adv: advanced search by the specific fields of title, creator (e.g. author or composer), subject, type (e.g. text or image), language, ISBN, or ISSN.
- view_brief: clicking on a library's collection to view its brief list of results.
- view_full: clicking on a title link in the list of brief records to expand it.
- col_set_theme: specifying a certain collection to search within.
- col_set_theme_country: specifying multiple collections to search or browse.

An important part of preprocessing the logs was session reconstruction. Each action was associated with a session ID and a timestamp. Actions of the same session were grouped together by session ID and then were sorted by timestamp. Details of the dataset and the preprocessing can be found in [1] and [2].

3 Analysis of Log File Entries

3.1 General Statistics

Table 1 presents statistics from the log analysis. Only a small proportion of the actions were performed by signed-in users (0.76%) compared to the number of actions recorded for guests (99.24%). This may indicate that users find it easier, and/or perhaps more secure, not to register in a web search system. Such behaviour sets a challenge to individualised personalisation.

Table 1. Descriptive Statistics

Item	Frequency
Actions by guests	1,619,587
Actions by logged-in users	12,457
Queries by guests	456,816
Queries by logged-in users	2,973
Sessions	194,627
User IDs	690

User actions were classified into four categories: *Search*, *Browse* (browsing or navigating result pages), *Collection* (limiting the search scope by selecting a collection or subject), and *Other*. Table 2 shows the distribution of actions along the categories. A considerable number of user actions (11.34%) were performed

before attempting the search, such as specifying collections for search. This indicates the diversity of user preferences, where users seek to customise the search environment according to their needs. User profiling may help to save user effort by automatically adjusting the search environment upon user identification.

Table 2. Broad classification of actions

Classification	Percentage
Search	28.17
Browse	56.79
Collection	11.34
Other	3.70

With regards to searching, it was found that there was a great inclination towards using simple search (16.14% of total actions) compared to using advanced search (4.35% of total actions). Another inclination was found in the pre-selection of a single collection for search, which occurred more frequently than the pre-selection of multiple collections (col_set_theme: 7.13% of actions; col_set_theme_country: 2.72%). This suggests that users seeking to limit their search tend to be very specific in selecting a collection. This may come from previous experience with the portal, where users found that certain collections had a higher degree of satisfying their information needs.

3.2 Query Reformulation

There are several types of reformulation of successive user queries: focusing on search terms and disregarding Boolean operators, a term can be added, deleted, or modified. For advanced search, in addition, a field can be added, deleted, or changed (some of the latter actions co-occur with operations on search terms). We defined four types of query reformulation, depending on the way query terms are affected: term addition, term deletion, term modification, and term change. Term modifications are changes to single-term queries.[2] No differentiation was made between queries submitted under different interface languages, because (i) the major part of the queries were submitted under English, and thus, the data for other interface languages might not be sufficient, and (ii) some query changes were manually observed as changing a query to another language.

As some users switch from the simple to the advanced search interface, related queries are difficult to identify if different types of queries are considered. For the following experiment, search terms were extracted from queries in order to identify how users typically modify a query. Only successive searches on the same topic were considered. To identify queries about the same topic, the following

[2] The distinction between term changes and term modifications originates from the definition of successive queries for queries with one and with more search terms.

approach was used: consecutive queries must have at least one term in common (if the query contains more than one search term) or a term in the query must have a Levenshtein distance [3] less than three to one in the other query. A query parser was implemented to extract the terms from the query log and identify the type of query modification and the most frequent changes.

Table 3 shows some of the reformulation classes based on the top 50 reformulations. A hyphen in Table 3 indicates operations which are not observable. It was found that 16% of term additions (add), 24% of term deletions (del), and 28% of term changes (chg) affected stopwords. Such changes might make sense under the assumption that users sometimes copy and paste text into a search box, and they might have just mistakenly inserted unwanted stopwords into the TEL search box. However, if the underlying indexing/retrieval system of TEL ignores stopwords, then adding or changing them will have no effect on search results, and would be considered a waste of effort for TEL users. A quick test reveals that stopword removal is handled inconsistently by the libraries in TEL, e.g. a search for *"the"* returns zero hits for the Austrian and French national library, but several thousand for the German and Belgian national library.

Table 3. Top 50 changes to terms in successive related queries

			Percentage			
Type	Brief description	Example	add	del	mod	chg
ST	use of stopwords	*"a"* → *"the"*	16	24	6	28
BL	use of Boolean operators	*"AND"* → *"OR"*	4	6	0	12
CC	change of lowercase or uppercase	*"europe"* → *"Europe"*	0	0	8	0
SC	spelling change	*"wolrd"* [!] → *"world"*	0	6	4	4
CH	use of special characters	*"*"* at the end of term	6	0	0	4
LC	language code change	*"ita"* → *"eng"*	2	2	0	20
RT	related terms	*"triangulum"* → *"quadratum"*	–	–	2	4
MO	morphologic variant	*"city"* → *"cities"*	–	–	26	2
TR	translation or transliteration	*"power"* → *"kraft"*	–	–	24	4
PN	change proper noun/name	*"mozart"* → *"amadeus"*	42	26	20	8
PI	single character (initials)	*"elzbieta"* → *"e"*	20	20	0	2
DT	date/number change	*"1915"* → *"1914"*	4	6	0	6
OT	unknown change/other	*"test"* → *"toto"*	6	10	10	6

Proper nouns and single characters (mostly denoting initials of names) made up 62% of term additions, 46% of deletions, 20% of modifications (mod), and 10% of changes. In contrast, term modification mostly affect morphological variations and translations (26% and 24%, respectively). Such modifications would not have any effect on the search results, because the TEL system does not seem to perform stemming.

Special characters (e.g. wildcards) were rarely used. Moreover, a small number of changes involved the use of semantically related terms (including narrower terms or broader terms). Also, only a small number of changes involved changing Boolean operators (e.g. *"AND" → "OR"*). This behaviour implies that some users are familiar with different search operators supported by the TEL portal.

The query reformulation analysis supports the hypothesis that a large group of users has little knowledge of the system, as they include stopwords and even change them (assuming TEL ignores stopwords as is commonly done by search engines). This group corresponds to novice users. On the other hand, a small group, corresponding to experienced users, used advanced query operators such as wildcards.

3.3 Interface Languages

In an attempt to investigate the relation between language and search behaviour, several variables were studied across the interface language selected by users of the portal. Actions were distributed among 30 languages. Hereafter, the study focuses on the top five languages in terms of the number of actions. The top language was English (86.47% of the actions), followed by French (3.44%), Polish (2.17%), German (1.48%), and Italian (1.39%). It is to be noted that the interface language does not necessarily imply the language of the query. One possible cause for the bias towards English, aside from its inherent popularity, is that it is the default language in the portal. Due to such anticipated bias, we will not include English (as an interface language, not as a query language) in further comparative discussions against other interface languages in this study. Nevertheless, we will show its associated percentages in subsequent tables for the sake of completeness. Possible ways to avoid this bias in the future would be to ask the user to specify a language before attempting the search, or to have the default language automatically specified according to the client's IP address.

The frequency distribution of the six main actions across the five languages is shown in Table 4. It was observed that users of the Polish language seemed to have a higher rate than others in using the feature of specifying a single collection before attempting the search. This finding may support the hypothesis that users from different linguistic or cultural backgrounds behave differently in search. However, we cannot rule out the fact that such observation may have been specifically governed by the amount of available collections in TEL.

3.4 Term Frequencies and Categories

As part of our analysis, the number of terms per query and the top queried terms for simple and advanced search were studied. Table 5 shows the mean and median of the number of terms per query across interface languages. It can be seen that German showed the lowest mean in both types of search. Moreover, part of the analysis revealed that German exhibited the largest distribution of queries made up of just one term. This may be because German noun compounds, which can express complex topics, are written as a single word.

Table 4. Action distribution across languages

Lang	search_sim	search_adv	view_brief	view_full	col_set_theme	col_set_theme_country
English	16.48%	4.32%	25.79%	30.65%	6.79%	2.66%
French	14.27%	4.46%	27.34%	23.55%	10.86%	3.12%
Polish	15.18%	4.23%	26.99%	21.95%	13.58%	3.39%
German	14.75%	4.31%	28.96%	23.53%	9.46%	2.93%
Italian	14.44%	6.16%	24.81%	28.39%	9.35%	2.78%

A comparison was made between the mean of the number of terms per query in simple search and the results reported in [4], which was a similar study applied on logs from AlltheWeb.com[3] (a European search engine that allows limiting the search to documents in a language of choice). With the exception of English, the means were approximately the same, despite the fact that the former is a library search system and the latter is a general search engine.

Table 5. Number of terms per query across interface languages

	Simple Search		Advanced Search	
Language	Mean	Median	Mean	Median
English	2.38	2	3.05	3
French	2.09	1	2.85	2
Polish	1.89	1	2.59	2
German	1.77	1	2.6	2
Italian	2.09	2	3.17	2

Part of the log analysis involved the extraction of the top 20 occurring search terms for each interface language, excluding stopwords. A term was only counted once in a session. This was done to avoid bias towards terms that were repeatedly searched for in the same session. Furthermore, terms were divided into five categories: *creator* (author, composer, artist, etc.), *location* (cities, countries, etc.), *subject* (as per Dewey Decimal Classification), *title* (including proper nouns and common nouns), and *type* (document types, e.g. text, image, sound). These categories were mostly based on the fields of the advanced search in TEL.

Figure 1 shows the category distribution of the top 20 search terms for each language. Differences were observed in user behaviour between different languages. For example, in simple search, 20% of the terms under French were subjects and 25% were creators, while under Italian, only 5% of the terms were subjects, while 40% of the terms were creators. Such findings reflect the differences between users of different languages and may contribute towards further research in multilingual query adaptation, perhaps suggesting a different adaptation strategy for each language or group of languages.

[3] http://www.alltheweb.com/

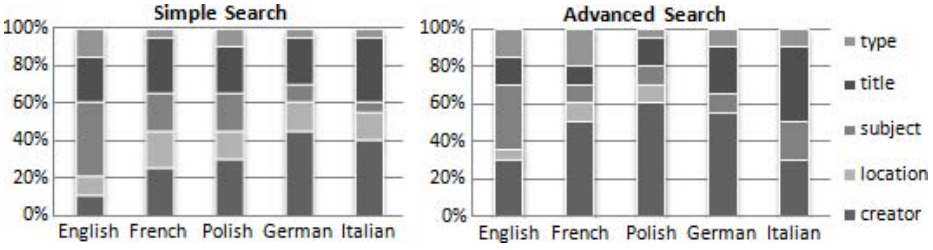


Fig. 1. Distribution of term categories across languages

3.5 Action Sequences

Table 6 shows patterns of two and three successive user actions. It points out the top most occurring patterns, as well as some other interesting patterns that have a high frequency. Related patterns are grouped together. It is observed that more users, after performing a search action, seem to directly view a full record (click for expansion) rather than clicking on a collection first (view_brief) before clicking to view full. The reason for this may be that the collection they wanted was already highlighted (TEL automatically highlights the top most collection in alphabetical order). This may indicate that more people prefer to specify collections before they perform the search so as to directly jump to view full without having to click on a collection.

Table 6. Selected sequential action patterns for two and three successive actions

Action 1	Action 2	Action 3	Frequency
view_full	view_full	–	153,952
search_sim	view_full	–	112,562
search_sim	view_brief	–	86,625
col_set_theme	search_sim	–	40,044
col_set_theme_country	search_sim	–	12,397
view_full	view_full	view_full	79,346
col_set_theme	col_set_theme_country	col_set_theme	4,735
col_set_theme_country	col_set_theme	search_sim	3,159

It can also be observed that users seem to get confused between two features (both accessible from TEL web site main page), which are: col_set_theme (choose a single collection) and col_set_theme_country (browse collections/choose multiple collections, which redirects the user to another page). This was observed as user actions subsequently alternated between the two features. Based on the pattern frequencies and the findings presented in Sect. 3.1, it can be inferred that

users prefer the feature of choosing a single collection. Perhaps deeper analysis of such patterns may introduce certain changes to the TEL portal's GUI.[4]

4 Summary and Outlook

We have described an analysis of multilingual search logs from TEL for the LADS task at CLEF 2009. The results of the analysis support the hypotheses that: (1) users from different linguistic or cultural backgrounds behave differently in search; (2) the identification of patterns in user actions could be useful for stereotypical grouping of users; and (3) user queries reflect the mental model or prior knowledge of a user about a search system.

The results suggest that there is scope for further investigation of how search logs can be exploited to improve cross-language search personalisation. Furthermore, the results imply that there is scope for improving the TEL system in a number of ways: (1) integrating a query adaptation process into TEL, where queries can be automatically adapted to retrieve more relevant results; (2) offering focused online help if a user spends an uncharacteristically long time between some actions or if a user performs a sequence of logically inconsistent actions; (3) highlighting elements in the TEL GUI as a default action or a typical next action; and (4) identifying the type of user for the sake of search personalisation.

Acknowledgments

This research is supported by the Science Foundation of Ireland (grant 07/CE/-I1142) as part of the Centre for Next Generation Localisation at Trinity College Dublin and Dublin City University (http://www.cngl.ie/).

References

1. Mandl, T., Agosti, M., Di Nunzio, G., Yeh, A., Mani, I., Doran, C., Schulz, J.M.: LogCLEF 2009: the CLEF 2009 Cross-Language Logfile Analysis Track Overview. In: Peters, C., Di Nunzio, G., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.) CLEF 2009. LNCS. Springer, Heidelberg (2010)
2. Ghorab, M., Leveling, J., Zhou, D., Jones, G., Wade, V.: TCD-DCU at LogCLEF 2009: An analysis of queries, actions, and interface languages. In: Borri, F., Nardi, A., Peters, C. (eds.) Results of the CLEF 2009 Cross-Language System Evaluation Campaign, Working Notes of the CLEF 2009 Workshop (2009)
3. Wagner, R.A., Lowrance, R.: An extension of the string-to-string correction problem. JACM 22(2), 177–183 (1975)
4. Jansen, B.J., Spink, A.: An analysis of web searching by European AlltheWeb.com users. Information Processing & Management 41(2), 361–381 (2005)

[4] Comments on the TEL GUI refer to the version accessible on the web in July 2009.

A Search Engine Based on Query Logs, and Search Log Analysis by Automatic Language Identification

Michael Oakes and Yan Xu

University of Sunderland, Dept. of Computing, Engineering and Technology, DGIC,
St. Peter's Campus, Sunderland SR6 0DD, England
{michael.oakes,yan.xu-1}@sunderland.ac.uk

Abstract. This work describes a variation on the traditional Information Retrieval paradigm, where instead of text documents being indexed according to their content, they are indexed according to the search terms previous users have used in finding them. We determine the effectiveness of this approach by indexing a sample of query logs from the European Library, and describe its usefulness for multilingual searching. In our analysis of the search logs, we determine the language of the past queries automatically, and annotate the search logs accordingly. From this information, we derive matrices to show that a) users tend to persist with the same query language throughout a query session, and b) submit queries in the same language as the interface they have selected.

1 Introduction

The hypothesis behind search log-based approaches to search engine design is that previous users' choices are of interest to new users who input similar queries [1] [2]. We take an extreme position on this, since rather than indexing documents based on their content as in conventional search engines, we index documents solely with the terms past users have used in searching for them, as found in the search logs. Collated under each downloaded document ID are the terms of every query ever submitted in a session leading up to the downloading of that document. Thus query terms from separate sessions leading to the retrieval of the same document will all become index terms for that document. This approach is beneficial for multilingual searching, since each previously downloaded document is indexed by all search terms which have ever been used in a search for that image, irrespective of which language they were in. Thus a document might be indexed by search terms in various languages, and therefore be accessible to queries in any of those languages. The limitation of this query log approach is that if previous users have never downloaded a particular document, then that document can never be retrieved by this technique. This means that there is a need for large amounts of search log training data, and the system is only suitable for indexing a closed corpus.

2 Implementation of a Search Engine Based on Query Logs

The experiments described in this paper were performed for the LADS (Log Analysis for Digital Societies) task in the LogCLEF track of CLEF in 2009 [3]. The first step in

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 526–533, 2010.

the analysis of query logs necessary to index search engine documents is to be able to identify the start and end of each query session. This is non-trivial in some data sets, but a unique ID is given to each session in the LogCLEF search logs. Query records for which the seventh field was "search_xxx", "available_at" or "see_online" were assumed to indicate the downloading of the relevant URL cited in field 12. These URLs would then be indexed by all the search terms submitted in either that session or any other session which resulted in the downloading of that same URL. The queries were not stop listed, since multilingual queries would require a stop list for each language and increase the danger that a stop word in one language might be meaningful in another. Instead, the tf-idf weighting scheme was used so that each query term would be given a weight reflecting its importance with respect to each document. In a conventional search engine, the highest tf-idf scores are given to words which occur frequently in the document of interest, but in few other documents. In the search engine based on query logs, the highest tf-idf scores are given to those terms which are often used in searches for the document of interest, but are not used in searching for many other documents. Once all the previously downloaded documents have been indexed, we can match the queries of future users against the document index terms using the cosine similarity coefficient, as in a conventional search engine. The documents are ranked according to their similarity to the user's query, and the best matching documents are presented to the user. The architecture of our query log-based search engine is shown in Figure 1.

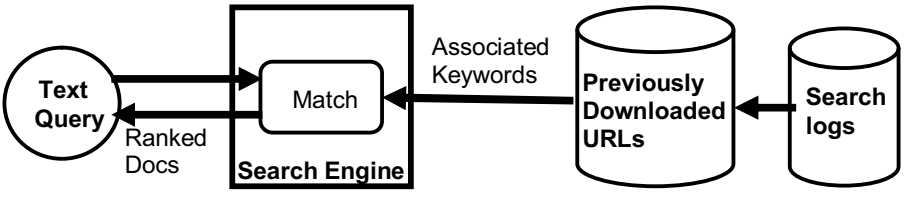


Fig. 1. Architecture of the Query Log-Based Search Engine

3 Evaluation of the Search Engine

For training the search engine we used 1399747 records (the first 75% when sorted chronologically) and for testing we used 466582 (the remaining 25%). Each record was a single line or row in the database of logs. It was necessary to sort the logs into chronological order in order to separate the test and the training set. If we define a search session as all records with a common session ID, the LogCLEF file contained 225376 sessions. Average session length according to this definition was 1399747 / 225376 = 6.21 lines. Some of these sessions recorded more than one retrieved URL.

For each URL in the test set, we combined all the query terms used in the same session into a single text query. We then matched this query against each of the query term sets collated under URLs in the test set. We assumed that the URL retrieved in the test session was the gold standard, and wished to determine the search engine's

ability to retrieve the session (if there was one) with the same URL from the training set. There were 8586 URLs in the test set sessions, and 284 of them matched previous records retrieving the same URL in the top 100 best matching training set sessions. Thus the percentage of matched URLs was 3.32%. The results appear much better when we consider that in the vast majority of cases, the gold standard URL could not be retrieved by our search engine, since it did not appear in the training set. When we considered only those 679 cases where the gold standard URL *could* have been retrieved, our results were as follows: On 2.9% of occasions, the gold standard URL was ranked first. On 13.7% of occasions, the gold standard URL was ranked in the top 10 retrieved documents, and on 34.0% of occasions it was ranked in the top 50.

4 Analysis of the Search Logs

The overall procedure we have followed for search log analysis is as follows: The search logs are read in, and we find the most likely language of the query terms on each line. Each line of the search log is then annotated with the name of the query language on that line, or "null" if no query is present, as described in section 5. The frequencies of each query language used in the first 100000 lines of the logs are given in Section 6. Given the sequence of query languages in the logs, we determine the likelihood of a query in one language (or a new session) being followed by a query in each of the other languages, another query in the same language, or the end of the session. This program is described in section 7. For each interface language, we determine the frequency of the query languages used, as described in section 8.

5 Automatic Language Identification

Souter et al. [4] describe a technique for automatic language identification, based on trigram (sequences of three adjacent characters) frequencies. Following this approach, we estimated the trigram frequencies typical of a set of languages from the Europarl corpus [5] which contains transcripts of meetings of the European Parliament in each of 11 languages. The languages of Europarl are Danish, Dutch, English, Finnish, French, German, Greek, Italian, Portuguese, Spanish and Swedish.

We counted the frequency of each trigram in a sample of just over one million characters for each language. The frequency of each trigram was divided by the total number of characters (minus two) in the sample of that language, to give the probabilility of that trigram being selected randomly from a text in that language, and stored in an external file. For example, the sequence "TZE" was found 37 times in the sample of 1058761 characters (1058759 trigrams) of German, giving a probability of occurrence of about 3.49 E -5 (3.49 times 10 to the power -5). To prevent trigrams which were not found in the million word sample being regarded as impossible in that language, we set a default value of 0.5 divided by the number of trigrams in the sample as the probability of that trigram. We automatically assigned initial and terminal blank characters to each query.

As an example, the overall probability of encountering the sequence _KATZE_ (where the underscore denotes a space character) was found by multiplying the five constituent trigram probabilities for each language (shown in Table 1), and multiplying them together to give an overall probability. Since the overall probability was much greater for German than the other two languages, KATZE is more likely to be a German word than an English or French one. Using this method, each of the first 100000 queries in the original search logs was annotated with the most likely language of that query.

Table 1. The five constituent trigrams in "_KATZE_", their individual probabilities and their product for three languages

Trigram	_KA	KAT	ATZ	TZE	ZE_	Overall Probability
English	1.23E-5	4.72E-7	4.72E-7	3.49E-5	1.70E-5	1.63 E-27
French	5.00E-7	5.00E-7	5.00E-6	5.00E-7	1.01E-5	6.31 E-32
German	5.64E-4	1.41E-4	1.41E-4	1.41E-4	1.12E-4	1.77 E-19

Other methods for automatic language identification include the text compression technique of Benedetto et al. [6]. An implementation of this technique in Perl is given by Goodman [7], but he reports that it performs less well and more slowly than Naïve Bayesian methods. A more common technique for automatic language identification is counting the number of language-specific stop words in each text, but this needs longer texts to work with.

The performance of our own automatic language identifier (ALI) was estimated by manually scoring the languages returned for the first hundred unique queries (ordered by the unique identifier in the first column of the search logs). This was a lengthy process, since many of the queries had to be checked on Google to shed more light on their language of origin. Closely related queries, such as "Johann Elias Reidinger", "JE Reidinger" and "Reidinger" were regarded as separate queries, whereas the eight occurrences of the query "Johann Reidinger" were considered a single query. The languages of proper nouns were marked according to the nationality of a person (and the language spoken there) or the current location of a place. Thus the name "Catharine Arley", which both sounds English and was identified as English by the system, was scored incorrect as a Google search showed that she was a French writer. Similarly, although the name of the town "Bischwiller" sounds German and was returned by the ALI as German, this was marked as incorrect as the town is now in France. Words which are used by more than one language, such as "aorta" (returned as Portugese), were marked as correct if the ALI identified one of the possible languages. Such decisions were verified using online translation systems. Queries such as "Rachmaninov" and "manga" which clearly do not belong to any of the languages of the ALI were marked incorrect. All the subjectivity inherent in this scoring method suggests the need for ground truth to be made available to the search log analysis community, consisting of search logs in which the language of each query, as determined by human expert annotators, is given.

As well as returning the most likely language of each query, the ALI also returns the second most probable language. The system was scored by the number of queries a) where the correct language was ranked first (47), and b) where the correct language was ranked either first or second (63). This left 37 queries where the correct language was not returned in either of the top two positions. The ALI had particular difficulty with the names of people, but performed best with queries consisting of three or more words. Of the 44 queries containing the names of people, on 18 occasions the correct language was deemed most likely, on 4 occasions it was deemed second most likely, and on 22 occasions the correct language was not selected. Of the 16 queries consisting of 3 or more words (including proper nouns), on 13 occasions the correct language was deemed most likely, twice it was deemed second most likely, and only once the correct language was not selected. It was also apparent from examination of the queries that users often have difficulty with spellings, trying several variants such as "Reidinger", "Riedinger" and "Ridinger". This suggests that the search engine would benefit from a spelling checker, like Google's "Did you mean?" facility, to be shared among all the language interfaces. The dictionary for such a spelling checker could be created from search log queries which had resulted in documents being downloaded, as these would probably be correctly spelled.

6 The Languages of Individual Queries

The automatic language identification program was used to estimate the languages of the first 100000 individual queries. Only the first 100000 queries were used, since the ALI took about one second per query. In 9.56% of the queries, no text was submitted. Among the queries where text was submitted, 29.69% were found to be in English, 13.38% in Italian, 12.27% in German, 9.56% in French, 7.84% in Dutch, 7.47% in Spanish, 5.61% in Finnish, 5.36% in Portuguese, 4.69% in Swedish and less than 0.01% in Greek.

7 Do Users Change Language within a Session?

The purpose of this experiment was to use the search logs annotated with the most likely language of the query to find whether users tended to stick with one language throughout a search session, or whether they tended to change languages in mid-session as part of the query reformulation process. The time-ordered set of query logs was scanned, and each time the session ID did not match the session ID of the previous query, it was assumed that a new query had begun. Otherwise, if the previous and current session IDs were the same, the entry [previous_state][current_state] in the matrix was incremented by 1. The results are shown in Table 2, where the earlier states ("from") are on the vertical axis, while the later states ("to") are found on the horizontal axis. "new" denotes the start or end of a session, and "null" indicates no query was submitted at this stage.

Table 2. Query language used in consecutive stages of query reformulation (Rows denote earlier query, Columns denote later query)

	new	null	De	En	Es	Fi	Fr	It	Nl	others
New	0	5990	5306	12901	3212	2384	4021	5633	3118	6083
Null	4838	3557	207	468	116	101	155	182	127	223
De	5462	48	5287	69	21	19	20	36	37	46
En	13259	142	63	12921	43	46	50	71	35	102
Es	3287	26	19	45	3218	11	27	37	20	36
Fi	2452	34	12	38	11	2412	17	24	13	38
Fr	4129	35	22	62	28	14	4224	38	22	31
It	5765	47	30	92	24	22	46	5916	23	42
Nl	3184	43	54	48	18	18	14	19	3619	39
Others	6271	53	45	88	35	24	31	51	42	6163

In the vast majority of cases, as shown by the high values on the principal diagonal, users having submitted a query in one language tended to use the same language for the next query. If users did change language mid-session, there was a slight tendency to change into English, shown by the slightly higher values in the "En" column. The other values in the matrix may represent a "noise floor" due to incorrect assignments by the automatic language identifier. A variant of this program was written to calculate the proportion of sessions in which more than one language was used. Here "null", or no query submitted, was not considered as a language. An example of this would be a query session where every line consisted of the selection of "col_set_theme" which did not require the submission of a text query. The relative proportions of sessions consisting of zero, one or more than one language were 4701 (9.66%), 42354 (87.07%) and 1592 (3.27%) respectively.

8 Correlation between the Portal Interface Language and the Query Language

The purpose of this experiment was to answer the question: Do users tend to submit queries in the same language as the interface they have chosen? First, the number of sessions conducted in each interface was collated, as shown in Table 3. The matrix in Table 4 was then generated by reading in each line of the annotated query logs in turn, reading off both the language of the interface and the language of the query and incrementing the entry [interface_language][query_language] by 1. For the most popular interfaces, German, French, Italian, Dutch, Portuguese and English, the most common query language was the language of the interface, followed by English. When the interface was English, the most common query language apart from English itself was Italian. This may in fact be due to fact that users were searching for documents with Latin titles. Latin is not included in the Europarl corpus, so our automatic language identifier in cases of Latin queries may have returned the most similar language, Italian. For example, the Latin query "Commentaria in Psalmos Davidicos" was returned as Italian. Another interesting finding was that in the sample of the search logs used by us, the Spanish interface was never selected, but many users wishing to submit Spanish queries used the interface of the closest language, Portuguese.

Table 3. Sessions grouped by language of the interface

Language code	Frequency	Percentage	Language code	Frequency	Percentage
En	41109	84.50	**Sr**	84	0.17
Pl	2012	4.14	**Sk**	79	0.16
Fr	1993	4.10	**Cs**	75	0.15
De	1145	2.35	**Nl**	55	0.11
It	511	1.05	**Lt**	41	0.08
Pt	353	0.73	**El**	28	0.06
Lv	346	0.71	**Fi**	13	0.03
Sl	340	0.70	**---**	10	0.02
Hu	245	0.50	**Da**	9	0.02
Hr	103	0.21	**Mt**	5	0.01
Et	91	0.19			

Table 4. Cross-tabulation for the choice of interface language (rows) and query submitted (columns)

	null	De	En	Es	Fr	It	Nl	Pt	Sk	others
Null	0	0	0	0	0	0	0	0	0	0
De	380	360	535	167	98	203	243	115	100	184
En	7856	9109	23892	5496	6985	10107	6014	4126	3650	7292
Es	0	0	0	0	0	0	0	0	0	0
Fr	561	320	789	265	1021	406	130	148	66	381
It	121	30	152	142	88	318	22	29	8	79
Nl	16	14	43	9	14	9	210	2	4	0
Pt	106	19	51	222	36	137	64	73	5	30
Sk	4	26	26	0	0	6	3	23	25	6
others	15	10	9	8	9	12	0	7	12	8

9 Conclusion and Future Work

We have developed a search engine, where previously accessed documents are indexed by all the search terms, derived from search logs, that have ever been submitted in the same sessions as those in which that document was downloaded. New queries are matched against the old query terms in the indexes, and documents are ranked by the degree of match between their index terms and the new query. This is a radically new approach, and initial results were encouraging.

In order to learn about multilingual searching behaviour, we have performed automatic language identification using trigram frequencies at the time the queries are indexed. We found that English was the most frequently selected query language, followed by "Italian", a figure which included the titles of many Ecclesiastical books written in Latin, erroneously classified as Italian by the automatic language identifier. Secondly, we found that users tend to submit queries in the same language as the interface they have selected, and also tend to stick with the same language throughout a query session. Due to the difficulties in judging the correctness of the language of each query, we feel that there is a need for ground truth judgments of the language of

a large number of queries, produced by human annotators, to facilitate the comparison of different automatic language identifiers. Examination of the variant spellings in the query logs arising within single sessions suggests that multilingual search engines would benefit from access to multilingual spelling checkers.

In future, we will build a hybrid search engine, where the degree of match between a query and document is the arithmetic mean of the match between the query and the document terms and the match between the query and the relevant search log terms. The arithmetic mean (rather than the geometric or harmonic mean) has been chosen so that even if one of the components returns a zero match, documents might still be found. This is to overcome the problem we encountered in this paper: when using search logs alone, we can only retrieve documents which have been retrieved in the past. This hybrid approach will be compared against the search engine developed in this paper, which uses search logs alone, and a traditional search engine which ranks documents by the similarity of their content to the current query. This comparison will be done using the same training/test set split that we have described in this paper. We will also repeat the experiments described in this paper using search logs recorded by the Exalead search engine [8] and BELGA, the Belgian news agency [9], who have recorded search logs for a very large searchable collection of commercially downloadable images with captions.

Acknowledgments. This work was supported by the EU-Funded VITALAS project (project number FP6-045389) http://vitalas.ercim.org

References

1. Hoi, C.-H., Lyu, M.R.: A Novel Log-Based Relevance Feedback Technique in Content-Based Image Retrieval. J. ACM Multimedia, 24–31 (2004)
2. Cui, H., Wen, J.-R., Nie, J.-Y., Ma, W.-Y.: Query Expansion by Mining User Logs. J. IEEE Transactions on Knowledge and Data Engineering 15(4), 829–839 (2003)
3. Mandl, T., Agosti, M., di Nunzio, G., Yeh, A., Mani, I., Doran, C., Schulz, J.M.: LogCLEF 2009: The CLEF 2009 Multilingual Logfile Analysis Track Overview. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece (2009)
4. Souter, C., Churcher, G., Hayes, J., Hughes, J., Johnson, S.: Natural Language Identification Using Corpus-Based Models. J. HERMES Journal of Linguistics 13, 183–203 (1994)
5. Europarl Parallel Corpus, `http://www.statmt.org/europarl`
6. Benedetto, D., Caglioti, E., Loreto, V.: Language Trees and Zipping. J. Physical Review Letters 88(4) (2002)
7. Goodman, J.: Extended Comment on "Language Trees and Zipping", `http://research.microsoft.com/en-us/um/people/joshuago/physicslongcomment.ps`
8. Exalead Search Engine, `http://www.exalead.com/search`
9. Belga News Agency, `http://www.belga.be/picture-home/index.html`

Identifying Geographical Entities in Users' Queries

Adrian Iftene

UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
adiftene@info.uaic.ro

Abstract. In 2009 we built a system in order to compete in the LAGI task (Log Analysis and Geographic Query Identification). The system uses an external resource built into GATE in combination with Wikipedia and Tumba in order to identify geographical entities in user's queries. The results obtained with and without Wikipedia resources are comparable. The main advantage of only using GATE resources is the improved run time. In the process of system evaluation we have identified the main problem of our approach: the system has insufficient external resources for the recognition of geographic entities.

1 Introduction

LogCLEF[1] deals with the analysis of queries as expression of user behavior. The goal of the task is the analysis and classification of queries in order to improve search systems [2]. Together with a group of students from the "Al. I. Cuza" University of Iasi, we competed in the task named *Log Analysis and Geographic Query Identification* (LAGI). In this task participants must create a system that is able to recognize geographic entities within a query stream. Competitors received two sets of logs: (1) one from Tumba! (a Portuguese web search engine) and (2) another from The European Library (TEL) multilingual information system.

The identification of geographic elements within a user query is the main aim for geographic information retrieval (GIR). The idea is that a geographical type question requires special treatment and, sometimes, a specific output oriented toward the specification of the geographical element (images, maps, geographical coordinates or landmarks).

The system that we built is based on the use of a geographical entity database. We built it starting from the English resources from GATE. We will show how the system was implemented and we will present which were the situations in which we did not correctly identify geographical entities in queries.

2 UAIC System for LAGI

In order to identify geographical entities, in addition to using the Wikipedia resources offered by the organizers, we built another resource containing geographical name entities starting from English resources from GATE.

[1] LogCLEF: http://www.uni-hildesheim.de/logclef/

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 534–537, 2010.

The **test data** consisted of two files, one from Tumba!, containing 152 entries, and one from TEL, with 108 entries; the aim was to add <*place*> </*place*> tags to user queries to incase geographical elements.

Resources: In order to build geographical entity resource, we started from GATE resources and additionally, we searched the web in order to add new similar entities. For different runs, we loaded either our own resources or the resources provided by the organizers (i.e. page titles from Portuguese and English versions of Wikipedia). From GATE [1] we used the following sets of named entities: *cities*, *countries*, *small regions*, *regions*, *mountains* and *provinces*.

The **main Module** loads the GATE or Wikipedia resources in a cache. For that it uses a hash map in which the key is the named entity itself, and the value is the number of words from initial named entity. Using this cache, our system tries to identify, in the TEL and Tumba test data, the geographical entities. The most important operations executed by the main module are: *resource loading*, *test data pre-processing* and *geographical entities identification*.

Resource Loading: Before loading geographical resources into the cache, a special method transforms all characters from these entities into lower case. Additionally, our program splits every name entity in component words and also loads these separated words in the cache, but specifies the number of words from the initial entity (in this way it will know if this key in our hash map comes from a simple name entity or from a composed name entity).

Test Data Pre-Processing: In order to identify geographical entities in the test data, the system performs pre-processing over it. The most important steps are parsing and identification of the user query, excluding of special characters and transforming of the query to lower case. The result is a new form of the user query, called *new query*.

Geographical Entity Identification is the most important operation of the main module. From now on the main question becomes: *How do we identify the geographical entities in this new query*?

Initially, we check whether the query itself is in the system cache. If we do, then the process of geographical entity identification is complete and we skip to the next line in test file. This is the case of the following line from TEL test data:

```
4752 & 11759 & ("portugal)"
```

for which the cache contains the new query *portugal* from the GATE list of countries.

If NO, then we try to split new query into single words, if possible. When we have only one word in the new query we automatically skip to the next line in test data file. When we have more than one word, we apply the following steps:

1. **Step 1** every individual word is searched in the hash map. If the current word comes from a simple named entity we simply add <place> tags to it. See line:

   ```
   4892 & 5670 & ("climbing on the Himalaya and other mountain ranges)"
   ```

   for which the cache contains the separated word *Himalaya* from the GATE *mountains* category.

2. **Step 2** for every word found in the hash map that comes from a compound named entity the system searches for it to the left and the right of the word, in order to combine more words with the same value in the hash map.
   2.1. If we have neighbors with the same attached value, then we create a common tag. This is the case of the following line:

```
13128 & 11516 & ("peter woods)"
```

   for which we have the keys *Peter* and *Woods* from GATE cities (first one from *Peter Tavy* and second one from *Harper Woods*). Because both have the same value in hash (2 which represents the number of words from initial entity) we create a common tag for both words.
   2.2. If we do not have neighbors with similar hash values, then we eliminate all these tags.

3 Submitted Runs

We submitted two sets of runs: one in which the main module loaded GATE as external resource, and one in which either the Portuguese or the English Wikipedia are loaded as external resources. The results are given in Table 1 (*Rcount* represents the *Reference count*, and *Hcount* represents *Hypothesis count*).

Table 1. UAIC Runs

Test Data	Resource	Rcount	Hcount	Match	Precision	Recall	F-measure
TEL	GATE	21	29	7	24.14	33.33	28.00
TEL	Wikipedia	21	99	16	16.16	76.19	26.67
Tumba	GATE	35	69	18	26.09	51.43	34.62
Tumba	Wikipedia	35	147	13	8.84	37.14	14.29

Comparing the results obtained based on the GATE derived resource and those obtained based on the Wikipedia derived resource, we observe that the former yields better precision while the latter better recall. Thus, using the GATE derived resource we identify a lower number of geographical entities compared to the Wikipedia derived resource (29 versus 99 for TEL test data, and 69 versus 147 for Tumba test data) even though the correct matches are comparable, and this is the reason for higher precisions in the former case. Regarding correct matches, in the case of TEL test data, the Wikipedia based system offered more correct matches and thus higher recall (76.19% versus 33.33%). For the Tumba test data the number of correct matches is higher for the GATE based system (18 versus 13 for the Wikipedia based system); this case yields the top F-measure of the system.

For the first run of the GATE based system on the TEL test data we have analyzed the results. First of all, we identified two cases where we offered partial results: we mark *"Belgium"* as a place together with double quotes and, for this reason, it was marked as incorrect, and from the geographical entity *mountain ranges* only marked *mountain*. From this analysis we deduce that one of the most important problems was

the fact that some geographical entities from the test data were not contained in the resources used. For example, the geographical resources from GATE lack the entities *Guttenberg*, *Peloponnese*, *Valetta*, *purnell*, *rus*, *Christiania* and *Arriaga*. For other entities, like *Wenlock Priory* we have parts of the name extracted from *Little Wenlock* and *Priory Wood* respectively, but we do not have the entire entity. An interesting case is the combination between *Gardner* and *London*, for which we have more than one entry in our list of cities, but both entities are not marked by our program. The mistake in this case comes from the fact that the application expects to entries of at most two words, and these entities have more than 3.

4 Conclusions

This paper presents the UAIC system which took part in the LogCLEF track in the LAGI task. The system uses external resources derived from GATE files and the two files offered by the organizers which were extracted from the Portuguese and English versions Wikipedia.

The main module is responsible for loading the external resources into the cache, for pre-processing the input text and for the identification of geographical entities. In the process of searching we use a cache mechanism to reduce run time.

The main problem of our system, and the most important direction of our future work, is related to the fact that the resources used were insufficient to identify all geographical entities from the test data. Other problems that we want to solve in the future were caused by the incorrect parsing of the user's query and by the incorrect handling of the cache containing geographical entities.

References

1. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: an architecture for development of robust HLT applications. In: ACL 2002: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 168–175. Association for Computational Linguistics (2001)
2. Mandl, T., Agosti, M., Di Nunzio, G.M., Yeh, A., Mani, I., Doran, C., Schulz, J.M.: LogCLEF 2009: the CLEF 2009 Multilingual Logfile Analysis Track Overview. In: Proceedings of the CLEF 2009 Workshop, Corfu, Greece, September 30 - October 2 (2009)

Search Path Visualization and Session Performance Evaluation with Log Files

Katrin Lamm, Thomas Mandl, and Ralph Koelle

Information Science, University of Hildesheim, Germany
{lammka,mandl,koelle}@uni-hildesheim.de

Abstract. This paper discusses new strategies for the performance evaluation of user search behavior. For the Log Analysis for Digital Societies (LADS) task of LogCLEF 2009 are proposed three different levels of user performance: success, failure and strong failure. The goal is to compare and measure session performance on a qualitative as well as on a quantitative level. The results obtained with both methods are in good agreement. Primarily they show that it is possible to investigate user performance from interpreting the user interactions recorded in log files. Both the qualitative and quantitative results give rise to a refinement of our operational definition of performance.

1 Introduction

This paper describes an approach to analyze logs from The European Library (TEL) within the LogCLEF track at the Cross Language Evaluation Forum (CLEF) 2009. Many different approaches have been applied to log file analysis [1]. We intend to identify deviations between usage behavior of users from different countries. These differences can be found in many aspects and currently it is unknown where the most relevant observation in this regard can be made. Consequently, we adopted an open approach, which allows the identification of differences in many areas and at various steps of the process. We believe that user path information could be a key to many findings.

Our approach uses the human visual capabilities to find trends and patterns by providing a visualization of user paths. We particularly decided to employ the hyperbolic tree view which provides a visualization of focus and context [2] and which has been applied for showing large hierarchies of data [3].

2 Definition of Performance

In our approach to analyze logs from TEL we assume that there are indicators, which suggest that a particular session was successful or not. One such indicator is when the user chose the *Available at Library* link to view the record in a particular national library interface. See [4] for a more detailed description of the individual user interactions recorded in the logs. In our judgment this action indicates that the user came across an interesting document according

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 538–543, 2010.

to the query and hence the session was successful. Within this paper our operational definition of performance includes the following three different levels of performance: A session is considered to be ...

- **successful** if one of the following actions is carried out at least once: available_at, see_online, option_print, option_save_reference, option_save_session_favorite, option_send_email, service
- **failed** if none of the actions above is carried out during the session
- **strongly failed** if none of the actions above is carried out and the user never uses the possibility of viewing a full record (view_full)

3 Search Path Visualization

We developed a log file analyzing algorithm and applied it to the provided search action log file. The algorithm is implemented in Java using JDBC as API to access the PostgreSQL database containing the log data. In the following we briefly describe our course of action. For a more detailed description see [5].

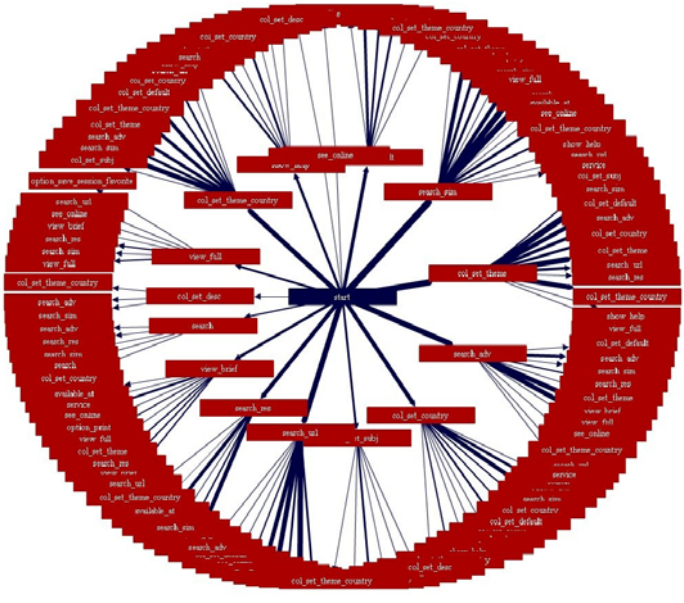


Fig. 1. Example search path visualization

The performance of each session was determined according to the definition of performance. For country recognition we integrated the *IP-to-Country Database*.[1] As the last two octets were missing in the logged IP addresses it was not possible to locate all users (24.11% could not clearly be assigned).

[1] The database is available at http://ip-to-country.webhosting.info/

To enable a more qualitative human assessment we visualized the sequence of individual user interactions with the interface of TEL. Therefore we generated several XML files following the GraphML file format[2] and adjusted an existing application for a simple social network visualization[3] for the interactive data visualization. An example search path visualization is shown in Fig. 1. In order to allow the analysis of multilingual search behavior we created visualizations for German, Spanish, French, British, Italian, Dutch, Polish and US users, as they could be identified by their IP addresses. The size of the edges of the search path graph indicates how often this search path could be observed within the logs. Aside from the total number of users that followed this path the visualization can be limited to only display successful, failed and strongly failed search paths. The user behavior can be retraced by navigating through the different search paths. Clicking on one of the actions causes the graph to rotate and then displays this action in the middle of the screen. The results as well as a more detailed description can be accessed online[4].

4 Analysis of User Path Information

Table 1 presents percentage values on the occurrence of sessions within the three levels of performance. The column entitled *total* refers to the number of sessions identified for the respective country. It can be seen from table 1 that, except for the Dutch and Polish results, the percentages do not vary much between the different countries. This already indicates that there may not be a lot of distinguishable differences between the different user groups.

Table 1. Occurrence of performance levels

	total abs.	success %	failure %	str. failure %
all	191781	13.00	56.08	30.92
British	7249	12.03	59.30	28.67
Dutch	6171	20.85	51.29	27.86
French	12574	16.09	51.62	32.29
German	9405	11.89	56.79	31.32
Italian	11979	12.56	57.48	29.96
Polish	7719	11.97	47.05	40.97
Spanish	12105	13.57	60.17	26.27
US	14953	12.22	57.83	29.94

Table 2 illustrates the first ten interactions of the most frequently used search path broken down into the three different levels of performance. Since we could not detect striking differences during the qualitative analysis of the user behavior for different countries at this point we only show the search paths for the group of all users. The success column contains missing values as after level 7 the most successful search path was ambiguous.

[2] http://graphml.graphdrawing.org/
[3] The application is available at http://flare.prefuse.org/download
[4] http://app01.iw.uni-hildesheim.de/logclef/visualizations.zip

Table 2. Most frequently used search paths

action	frequency	success	failure	str. failure
1	search_sim	search_sim	search_sim	search_sim
2	view_full	view_full	view_full	view_brief
3	view_full	view_full	view_full	search_sim
4	view_full	available_at	view_full	view_brief
5	view_full	search_sim	view_full	search_sim
6	view_full	view_full	view_full	view_brief
7	view_full	view_full	view_full	search_sim
8	view_full	-	view_full	view_brief
9	view_full	-	view_full	search_sim
10	view_full	-	view_full	view_brief

Our first observation is that the most frequently used and the most unsuccessful search path are identically for six of the nine user groups (British, Dutch and Polish users deviate). If we do not assume that most of the sessions (see also table 1) and the most frequently used search path are not successful, we might have to rethink our operational definition of failure. Maybe there are users that are already satisfied by having the possibility to view a full record (view_full) and maybe some TEL users use the library primarily for informative reasons.

As to the most frequently used search path we identified two different search patterns. British, Dutch, Italian and Spanish users act like we have seen before in the case of all users (see table 2). They submit a query and then view the results. German, French and US users submit a second query after viewing two full records and a third query after viewing again two full records. A possible interpretation of these differences would be that we deal with two types of users here. The first type prefers to sift through the list of search results; he submits his query and then examines at least eight documents without rephrasing his query once. The second type, however, prefers rephrasing his queries subsequently; he only examines a few documents of the result list in detail before rephrasing. As we do not know what results the users viewed it stays open whether the first type submits more eloquent queries that return better results or whether the second type is more aware of relevant documents according to his query. Table 3 illustrates these two different search behavior patterns.

Table 3. Two different search patterns

action	pattern A	pattern B
1	**search_sim**	**search_sim**
2	view_full	view_full
3	view_full	view_full
4	view_full	**search_sim**
5	view_full	view_full
6	view_full	view_full
7	view_full	**search_sim**
8	view_full	view_full
9	view_full	view_full
10	view_full	**search_sim**

As we suspected that pattern B represents a process of substantiating the initial query, in a further analysis we compared the first four queries of French, German and US sessions that follow the most frequently used search path of these countries. Surprisingly users of the second pattern in the majority of cases do not rephrase their queries while they perform a simple search. Table 4 depicts the number of sessions found where users actually reformulate their queries. We examined sessions with a minimum of 4, 7 and 10 first interactions in accordance with the most frequently used search path ($\geq$ 4, 7 or 10). Total refers to the number of sessions available with at least 4, 7 or 10 interactions. As can be seen from table 4 altogether the number of surveyed sessions is rather small in order to make reliable statements. Within the first four or more interactions of the most frequently used search path we found 2 German, 11 French and 11 US sessions with one query reformulation and there was never a second or a third reformulation. We suspect that this might be due to some technical reason in connection with the logging of user interactions. Maybe there are several actions recorded as *search_sim* although the users actually do not perform a new search.

Table 4. Sessions with query reformulation

	German			French			US		
$\geq$	**4**	**7**	**10**	**4**	**7**	**10**	**4**	**7**	**10**
total	49	6	2	70	8	6	93	9	3
1	2	1	1	11	2	1	11	1	1
2	-	0	0	-	0	0	-	0	0
3	-	-	0	-	-	0	-	-	0

5 Relative Frequencies of Performance Levels

With respect to quantitative results we calculated the relative frequencies of the three levels of performance depending on the number of interactions with the system as well as on their duration (for further results and figures see [5]).

As expected the success probability increases with the number of interactions, while the probability of strong failure decreases. Surprisingly the fraction of failed sessions is from 5 up to 40 interactions per session almost constant at 0.6%. This effect again points to the fact that our operational definition might not capture the full picture. As noted earlier some of the 0.6% TEL users might indeed be satisfied by using the library primarily for informative reasons, e.g. by reading the full records. In our opinion this long session durations indicate successful sessions, because otherwise the users would have abandoned their search earlier.

In order to analyze the influence of the session duration we grouped the sessions in five minutes intervals, i.e. from 0 up to 5 minutes, from longer than 5 up to 10 minutes etc. It can be seen from the data that after the first 20 minutes the proportions of success, failure and strong failure stay approximately constant. In other words, after the first 20 minutes the probability of a successful search becomes independent of the session duration. This could reflect the differences in the search speed of different users.

6 Conclusion and Outlook

The aim of this study was to experiment with new methods for log file analysis. As a starting point we developed an operational definition of search performance with three different levels. To enable a more qualitative human assessment we visualized the sequences of individual user interactions with the interface of TEL. Both, the more qualitative analysis of the search path visualizations as well as the more quantitative analysis of the logs have shown inconsistencies within the data which suggest that our operational definition of performance should be adjusted to include a fraction of the formerly failed sessions in the successful ones. The problem remains to identify appropriate indicators for this distinction. One proposal for an additional success indicator for future research is whether the session ends with a search or not. This could imply that the user did not find the information he/she was looking for and therefore the session ought to be evaluated as not successful. During the qualitative analysis of the user path information we observed some differences between users from different countries, e.g. that there seem to exist two prevailing search patterns. An interpretation in terms of query reformulation is not supported by the data, neither do the logs suggest another apparent explanation.

Although further research is needed to confirm our findings, basically we can say now that it is possible to investigate user performance from log files and that our refined definition of performance at least in the context of TEL users accounts for the user behavior.

References

1. Jansen, B., Spink, A., Taksa, I.: Handbook of Research on Web Log Analysis. IGI Global, Hershey (2009)
2. Lamping, J., Rao, R., Pirolli, P.: A focus+content technique based on hyperbolic geometry for viewing large hierarchies. In: CHI 1995: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 401–408. ACM Press, New York (1995)
3. Heo, M., Hirtle, S.C.: An empirical comparison of visualization tools to assist information retrieval on the web. Journal of the American Society for Information Science and Technology (JASIST) 52(8), 666–675 (2001)
4. Mandl, T., Agosti, M., Di Nunzio, G., Yeh, A., Mani, I., Doran, C., Schulz, J.M.: LogCLEF 2009: the CLEF 2009 Cross-Language Logfile Analysis Track Overview. In: Peters, C., et al. (eds.) CLEF 2009 Workshop, Part I. LNCS, vol. 6241, pp. 506–515. Springer, Heidelberg (2009)
5. Lamm, K., Koelle, R., Mandl, T.: Search Path Visualization and Session Success Evaluation with Log Files from The European Library (TEL). In: Peters, C., et al. (eds.) Working Notes: 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece (2009), http://www.clef-campaign.org/2009/working_notes/lamm-paperCLEF2009.pdf

User Logs as a Means to Enrich and Refine Translation Dictionaries

Alessio Bosca and Luca Dini

CELI s.r.l., 10131 Torino, Italy
{alessio.bosca,dini}@celi.it

Abstract. This paper presents the participation of the CACAO prototype to the Log Analysis for Digital Societies (LADS) task of LogCLEF 2009 track. In our experiment we investigated the possibility to exploit the TEL logs data as a source for inferring new translations, thus enriching already existing translation dictionaries. The proposed approach is based on the assumption that in the context of a multilingual digital library the same query is likely to be repeated across different languages. We applied our approach to the logs from TEL and the results obtained are quite promising.

1 Introduction

The Log Analysis for Digital Society (LADS) from LogCLEF track is a new task that focuses on the log analysis as a means to infer new knowledge from user logs (i.e. users behaviors, multilingual resources). In particular the task proposes to the participants to deal with logs from The European Library (TEL); [1] provides an overview of the data proposed for the task.

In the last years, starting from the so-called Web 2.0 revolution, the academic and research community showed an increasing interest towards the analysis of user generated contents (blogs, forums and collaborative environments like Wikipedia) in order to exploit this large information source and infer new knowledge from it (see [2] or [3]). Moreover, explicit user contributions are increasingly integrated in very specific, task-dependent activities like query disambiguation and translation refinements (i.e. the 'Contribute a better translation' strategy in Google Translate services, see [4]); such trend underlines that capitalizing user generated data is a key challenge in tuning and tailoring search system performances to real users needs.

In such a broader research context, significant efforts have focused on the analysis of data stored in transaction logs of Web search engines, Intranets, and Web sites as a means to provide a valuable insight for understanding how search engines are used and the users interests and query formulation patterns. These efforts are directed towards specific goals like inferring the search intents of users, identifying user categories through their search patterns and facilitating the personalization of contents or inferring semantic concepts or relations by clustering user queries.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 544–551, 2010.

Jansen [5]) presents a review on the Web search transaction log analysis and constitute a foundation paper for the research on this domain, while several works investigates on more specific, task related, topics. A paper from Wang and Zhai (see [6]) proposes to mine search engine logs for individuating patterns at the level of terms through the analysis of terms's relations inside a query in order to support and enhance query refinement. The authors of [7] investigate the use of query logs in improving Automatic Speech Recognition language models for a voice search application; Andrejko [8] describes an approach to user characteristics acquisition based on automatic analysis of user behavior within query logs thus minimizing the amount of necessary user involvement for personalization purposes.

CACAO (Cross-language Access to Catalogues And On-line libraries) is an EU project devoted to enabling cross-language access to the contents of a federation of digital libraries with a set of software services for harvesting, indexing and searching over such data.

In our experiment we focused on the multilingual aspect of log analysis and in particular we investigated the possibility to exploit the TEL logs data as a source for inferring new translations, thus enriching already existing translation resources for dictionary based cross language access to digital libraries. The proposed approach is based on the assumption that when users are aware of consulting a multilingual digital collection, they are likely to repeat the same query several times, in several languages. Such assumption can be illustrated with two different scenarios:

- In the first case the same user, that has a good knowledge of different languages, actively repeats the same query in different languages in order to increase the number of relevant documents retrieved; for instance, a first search can be performed in his own native language (i.e. French) and then repeated in English because he is not satisfied by the results retrieved with the first query.
- In the second case we can have two different and unrelated users, using natively different languages and casually searching for the same information, since they are consulting the same collection and queries tend to be topic-convergent in specific domains (see [9]).

By adopting the proposed algorithm over the previously described scenarios, it is possible to discover translationally equivalent queries in logs by monitoring user queries. The equivalent query pairs extracted can then be exploited in order to increase the coverage of translation dictionaries as well as providing new contextual information for disambiguating translations.

This paper is organized as follows: section 2 presents the architecture of our system, while in section 3 experiments and investigations on the Logs data are described and the obtained results presented; we finally conclude in section 4.

2 CACAO Project

CACAO (Cross-language Access to Catalogues And On-line libraries) is an EU project funded under the eContentplus program and proposes an innovative approach for accessing, understanding and navigating multilingual textual content in digital libraries and OPACs, thus enabling European users to better exploit the available European electronic content (see [10]).

CACAO project proposes a system based on the assumptions that users look more and more at library contents using free keyword queries (as those used with a web search engine) rather than more traditional library-oriented access (e.g. via Subject Heading); therefore, the only way to face the cross-language issue is by translating the query into all languages covered by the library/collection (rather than, for instance, translating subject headings, as in the MACS approach, https://macs.vub.ac.be/pub/). The system will then yield results in all desired languages; a prototype of the system is available on line at http://www.cross-library.com.

The general architecture of the Cacao system could be summarized as the result of the interactions of few functional subsystems, coordinated by a central manager and reacting to external stimuli represented by end users queries:

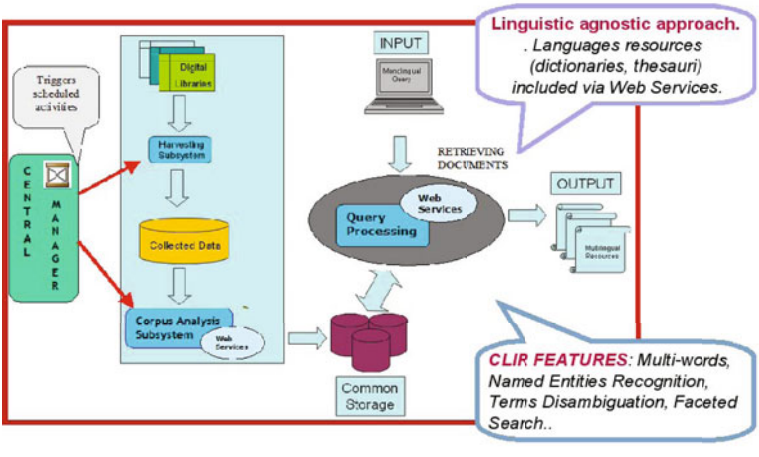


Fig. 1. CACAO System Architecture

2.1 Query Processing Subsystem Enhancement through User Logs Analysis

The CACAO system, as translation strategy, preferred to exploit bilingual dictionaries over machine translation (MT) technology for the following reasons:

– Typical MT systems under-perform in syntactically poor contexts such as web queries.

- With ambiguous translations most MT systems make a choice, thus limiting the retrieval of potentially interesting digital items
- Dictionary based techniques allow for search of (possibly disambiguated) multiple candidate translations.
- Dictionaries can be easily updated by personnel of the library to remedy specific lacks of coverage or for proposing translations that are more pertinent to the domain.

The critical issues that lie within the approach based on a bilingual dictionary consist in dictionaries coverage and translation candidate disambiguation. The issue of dictionary coverage originates from the absence of required terms from the dictionary and a strategy for retrieving such resources (exploiting user suggestions, inferring new translations from user log analysis, manual enrichment of dictionary by personnel of the library/system). The issue of disambiguation originates from polisemic terms whose translations bear different meaning (i.e. bank could be translated in Italian as "banca" or "sponda", with the first term related to economy and the second to rivers).

In such a context the analysis of queries from user logs could prove to be a valuable resource in order to enhance CACAO Query Processing Subsystem in terms of

- Detecting terms used in the queries and not present in the bilingual dictionaries
- Enriching already existing translation resources with new translation pairs
- Learn new Named Entities (this specific aspect is not treated in the present work)

3 TEL Logs Analysis

In our experiments we investigated the opportunity to exploit the TEL logs data as a source for inferring new translations. We focused on the identification of sets of user queries within the TEL logs, expressed in different languages, that are potential mutual translations and on their evaluation in order to retain only the eligible candidates. The experiments were designed according to the hypothesis described in the introduction: users that are consulting a multilingual digital collection are likely to implicitly and explicitly repeat the same query in different languages. A first task targeted the identification of queries explicitly repeated by users aware of the multilinguality of the underlying collection and with a working knowledge of the different languages, while a second one focused on the detection of translations candidates implicitly expressed by different users and presents in the logs. The second task consisted in the creation of a search index and in the use of the CACAO Cross Language Information Retrieval system in order to detect the potential candidate translations. The approach proposed in this paper, named the T-Like algorithm, allows to evaluate the candidate translations identified in the previous tasks by measuring the probability for two queries to be one the translation of the other; thus detecting the translationally equivalent query pairs among the candidate ones.

3.1 Explicit Query Repetition

The experiment consisted in the detection of a list of successive queries submitted in different languages by the same user. For each query *Q0* present in the logs, it has been retrieved a list of queries (*Q1*, *Q2*, .., *QN*) submitted by the same user (identified by means of information on session ID and user IP as well present in the logs) in a different language within 5 minutes from the submission time of *Q0*; the language of the query has been identified by means of a language guesser. The result of this task consisted in a list of translation candidates.

3.2 TEL Logs Search Index

This experiment consisted in the creation of a Lucene Search index (see [11]) starting from the TEL logs; information contained in the query field of the logs has been filtered in order to remove terms pertaining to the query syntax (restrictions on fields, boolean operators,...) and it has been enriched by means of shallow NLP techniques as lemmatization and named entities recognition. A language guesser facility was used in order to identify the query source language.

A second step involved the CACAO search engine in order to create a resource containing all possible translation candidates sets; each distinct query contained in the logs has been used as input for the CACAO system in order to obtain a set of translation for the query. The CACAO system translated the query from the TEL logs into all the languages it natively supports (English, French, German, polish, Hungarian and Italian) and then it exploited such translations in order to search for related queries in other languages; the result of this task consisted in a list of translation candidates proposed by the CACAO engine.

3.3 T-Like Algorithm

The last step of our investigations consisted in applying the T-Like procedure in order to evaluate the probabilities associated to the different translations candidates that were extracted from the logs with the previous task and thus obtain a list of proposed translations.

The TLike algorithm depends on three main resources:

- A system for Natural Language Processing able to perform for each relevant language basic tasks such as part of speech disambiguation, lemmatization and named entity recognition, we adopted the well known Incremental Parser from Xerox (see [12]).
- A set of word based bilingual translation modules following the approach ‘one source, many targets.‘.
- A semantic component able to associate a semantic vectorial representation to words.
  - dog [0.2 0.1 0 0 0 0 0.4 .] (semantic vector)

The basic idea beyond the TLike algorithm is to detect the probability for two queries to be one a translation of the other; given 2 queries *Q1,Q2* for each word (*wi*) in the query *Q1*:

- Obtain translations of *Q1* in language *T*.
- If translations exist
  - Verify if there is a word in *Q2* that is contained in the translations
  - If it exists, set a positive score and mark the word as consumed.
  - If intersection is empty, set a negative score and continue
- If translations do not exist or the intersection is empty:
  - Obtain the neighbors vector of *wi* *Vs*
  - Obtain a vector of translations of *Vs* *Vt*
  - Obtain a set of semantic vectors associated to each word in *Vt* *St*
  - Compose *St* to obtain a single semantic vector *Vc*.
  - For each word *wj* in *Q2*:
    - Compute the cosine distance between *Vc* and the semantic vector of that *wj*
    - If the distance is lower than a certain threshold accept the possible candidate
    - Continue until optimal candidate is found

Overall, the computational costs of the proposed algorithm can be quite high since it requires, as a prerequisite, the construction of specific corpus-based resources (the semantic vectors and the search index) and the computational costs involved in their generation directly depend on the size of the log corpus itself. However, in the scenario we presented, our application aims at enriching translation resources and the proposed approach is intended as an off-line procedure without specific constraints on the performance.

3.4 Experiments Results

Table 1 presents some statistic measures on the translationally equivalent groups retrieved by the system. The table reports a human estimation of a set of translation groups randomly chosen from the output of the T-Like algorithm: 100 among the translation groups evaluated as eligible candidates and 100 among the discarded ones. The results presented in the table concerns this evaluation tests and differs from the ones presented at the CLEF2009 workshop (see [14])) as they summarized the outcome of the automatic evaluations process performed by the system.

Looking at the results presented in the table we can observe that the false positive rate is higher than the false negative one. This behavior depends on the value of the similarity threshold used to prune away uncorrect translation candidates since we decided to privilege the precision over the recall in order to enrich the original translation resources with reliable data.

Table 2 instead presents an excerpt of the translation pairs extracted from the TEL logs and evaluated as eligible candidates by the T-Like algorithm.

Table 1. Evaluation Measures

true Positive translations	77
true Negative translations	93
false Positive translations	23
false Negative translations	7

Table 2. Examples of Query Pairs extracted from TEL Logs

Source Query in Logs	Candidate Translations from Logs
the road to glory [en]	en route pour la gloire [fr]
la vita di gesu narrata [it]	essai sur la vie de jsus [fr]
die russische sprache der gegenwart [de]	russian language composition and exercises [en]
digital image processing [en]	cours de traitement numrique de l' image [fr]
biblia krolowej zofii [pl]	simbolis in the bible [en]
national library of norway [en]	biblioteka narodowa [pl]
la guerre et la paix [fr]	war+and+peace [en]
storia della chiesa [it]	church history [en]
firmen landwirtschaftliche maschinen [de]	l'agriculture et les machines agricoles [fr]
dictionnaire biographique [fr]	dizionario biografico [it]
deutsche mythologie [de]	the mythology of aryan nations [en]
ancient maps [en]	carte antique [fr]
around the world in 80 [en]	le tour du monde en 80 [fr]

4 Conclusions

This paper represents the first step of a research on NLP based query log analysis. The translation equivalent pairs identified with the T-Like algorithm and exemplified in table 2 can be exploited in order to enrich our translation system by adding new translations into the bilingual dictionaries or by defining specific translation contexts for terms already present in the translation resources thus supporting the disambiguation strategies with a corpus of translation equivalent query pairs.

The preliminary results are quite encouraging and in the future we plan to extend this research in order to:

- learn Named Entities not present in the translation resources
- extend the semantic matching method to cover cases where the semantic vectors are not present in the semantic repository. This will imply the use of the web and web search engines as a dynamic corpus(on this topic see [15]).

Acknowledgements

This work has been supported and founded by CACAO EU project (ECP 2006 DILI 510035).

References

1. Hofmann, K., de Rijke, M., Huurnink, B., Meij, E.: A Semantic Perspective on Query Log Analysis Working Notes for the CLEF 2009 Workshop (2009)
2. Zaragoza, H.: Search and Content Analysis in the Web 2.0 Invited Talk at the Search and Content Analysis in the Web 2.0. In: WWW 2009 Workshop (2009)
3. Jijkoun, V., Khalid, M.A., Marx, M., de Rijke, M.: Named Entity Normalization in User Generated Content. In: AND 2008: Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data (2008)
4. Google Translate, http://translate.google.com/
5. Jansen, B.J.: Search log analysis: What it is, what's been done, how to do it. Library & Information Science Research 28(3), 407–432 (2006)
6. Wang, X., Zhai, C.: Mining term association patterns from search logs for effective query reformulation. In: CIKM 2008: Proceeding of the 17th ACM Conference on Information and Knowledge Mining, pp. 479-488 (2008)
7. Li, X., Nguyen, P., Zweig, G., Bohus, D.: Leveraging multiple query logs to improve language models for spoken query recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (2009)
8. Andrejko, A., Barla, M., Bielikov, M., Tvaroek, M.: User characteristics acquisition from logs with semantics. In: ISIM 2007 Information Systems and Formal Models 10th International Conference on Information System Implementation and Modeling, pp. 103–110 (2007)
9. Bosca, A., Dini, L.: The role of logs in improving cross language access in digital libaries. In: Proceedings of the International Conference on Semantic Web and Digital Libraries (2009)
10. Bosca, A., Dini, L.: Query expansion via library classification systems. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 42–49. Springer, Heidelberg (2009)
11. Lucene. The Lucene search engine, http://jakarta.apache.org/lucene/
12. At-Mokhtar, S., Chanod, J.-P., Roux, C.: Robustness beyond shallowness: incremental dependency parsing NLE Journal (2002)
13. Sahlgren, M.: An Introduction to Random Indexing. In: Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, Copenhagen, Denmark (2005)
14. Bosca, A., Dini, L.: CACAO Project at the LogCLEF Track. In: the Working Notes of Log File Analysis (LogCLEF) Track at CLEF 2009 (2009)
15. Baroni, M., Bisi, S.: Using co-occurrence statistics and the web to discover synonyms in technical languag. In: Proceedings of 4th International Conference on Language Resources and Evaluation (LREC), pp. 1725–1728 (2004)

CLEF 2009: Grid@CLEF Pilot Track Overview

Nicola Ferro[1] and Donna Harman[2]

[1] Department of Information Engineering, University of Padua, Italy
ferro@dei.unipd.it
[2] National Institute of Standards and Technology (NIST), USA
donna.harman@nist.gov

Abstract. The Grid@CLEF track is a long term activity with the aim of running a series of systematic experiments in order to improve the comprehension of MLIA systems and gain an exhaustive picture of their behaviour with respect to languages.

In particular, Grid@CLEF 2009 is a pilot track that has started to move the first steps in this direction by giving the participants the possibility of getting experienced with the new way of carrying out experimentation that is needed in Grid@CLEF to test all the different combinations of IR components and languages. Grid@CLEF 2009 offered traditional monolingual ad-hoc tasks in 5 different languages (Dutch, English, French, German, and Italian) which make use of consolidated and very well known collections from CLEF 2001 and 2002 and used a set of 84 topics.

Participants had to conduct experiments according to the CIRCO framework, an XML-based protocol which allows for a distributed, loosely-coupled, and asynchronous experimental evaluation of IR systems. We provided a Java library which can be exploited to implement CIRCO and an example implementation with the Lucene IR system.

The participation has been especially challenging also for the size of the XML files generated by CIRCO, which can become 50-60 times the size of the collection. Of the 9 initially subscribed participants, only 2 were able to submit runs in time and we received a total of 18 runs in 3 languages (English, French, and German) out of the 5 offered. The two participants used different IR systems or combination of them, namely Lucene, Terrier, and Cheshire II.

1 Introduction

Much of the effort of *Cross-Language Evaluation Forum (CLEF)* over the years has been devoted to the investigation of key questions such as "What is *Cross Language Information Retrieval (CLIR)*?", "What areas should it cover?" and "What resources, tools and technologies are needed?" In this respect, the Ad Hoc track has always been considered as the core track in CLEF and it has been the starting point for many groups as they begin to be interested in developing functionality for the multilingual information access. Thanks to this pioneering work, CLEF produced, over the years, the necessary groundwork and foundations

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 552–565, 2010.

to be able, today, to start wondering how to go deeper and to address even more challenging issues [11,12].

The Grid@CLEF Pilot track[1] moves the first steps in this direction and aims at [10]:

- looking at differences across a wide set of languages;
- identifying best practices for each language;
- helping other countries to develop their expertise in the *Information Retrieval (IR)* field and create IR groups;
- providing a repository, in which all the information and knowledge derived from the experiments undertaken can be managed and made available via the *Distributed Information Retrieval Evaluation Campaign Tool (DIRECT)* system.

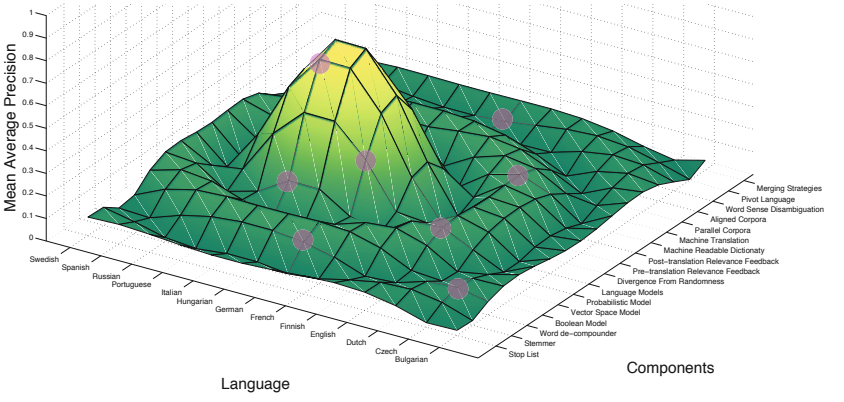
The Grid@CLEF pilot track in CLEF 2009 has provided us with an opportunity to begin to set up a suitable framework in order to carry out a first set of experiments which allows us to acquire an initial set of measurements and to start to explore the interaction among IR components and languages. This initial knowledge will allow us to tune the overall protocol and framework, to understand what directions are more promising, and to scale the experiments up to a finer-grain comprehension of the behaviour of IR components across languages.

The paper is organized as follows: Section 2 provides an overview of the approach and the issues that need to be faced in Grid@CLEF; Section 3 introduces CIRCO, the framework we are developing in order to enable the Grid@CLEF experiments; Section 4 describes the experimental setup that has been adopted for Grid@CLEF 2009; Section 5 presents the main outcomes of this year Grid@CLEF in terms of participation and performances achieved; finally, Section 6 discusses the different approached and findings of the participants in Grid@CLEF.

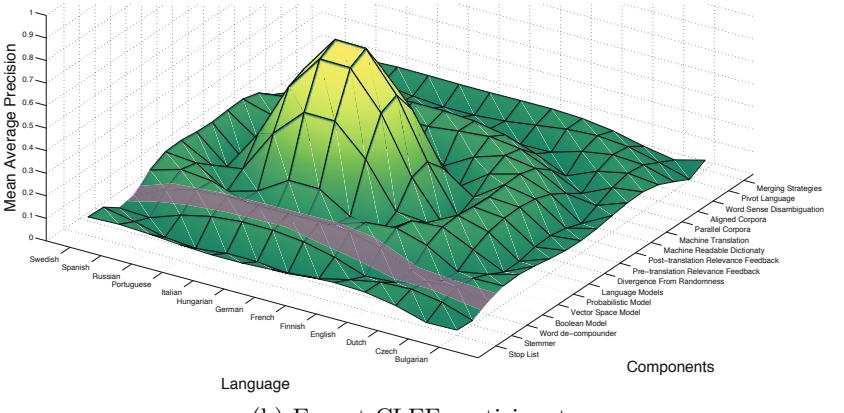
2 Grid@CLEF Approach

Individual researchers or small groups do not usually have the possibility of running large-scale and systematic experiments over a large set of experimental collections and resources. Figure 1 depicts the performances, e.g. mean average precision, of the composition of different IR components across a set of languages as a kind of surface area which we intend to explore with our experiment. The average CLEF participants, shown in Figure 1(a), may only be able to sample a few points on this surface since, for example, they usually test just a few variations of their own or customary IR model with a stemmer for two or three languages. Instead, the expert CLEF participant, represented in Figure 1(b), may have the expertise and competence to test all the possible variations of a given component across a set of languages, as [22] does for stemmers, thus investigating a good slice of the surface area.

[1] http://ims.dei.unipd.it/gridclef/



(a) Average CLEF participants.



(b) Expert CLEF participant.

Fig. 1. Coverage achieved by different kinds of participants

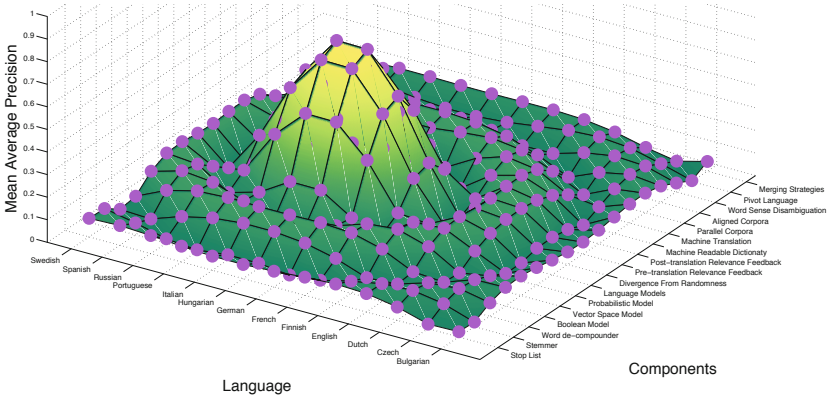


Fig. 2. The three main entities involved in grid experiments

However, even though each of these cases produces valuable research results and contributes to the advancement of the discipline, they are both still far removed from a clear and complete comprehension of the features and properties of the surface. A far deeper sampling would be needed for this, as shown in Figure 2: in this sense, Grid@CLEF will create a fine-grained *grid of points* over this surface and, hence, the name of the track comes.

It is our hypothesis that a series of systematic experiments can re-use and exploit the valuable resources and experimental collections made available by CLEF in order to gain more insights about the effectiveness of, for example, the various weighting schemes and retrieval techniques with respect to the languages.

In order to do this, we must deal with the interaction of three main entities:

- **Component:** in charge of carrying out one of the steps of the IR process;
- **Language:** will affect the performance and behaviour of the different components of an *Information Retrieval System (IRS)* depending on its specific features, e.g. alphabet, morphology, syntax, and so on.
- **Task:** will impact on the performances of IRS components according to its distinctive characteristics;

We assume that the contributions of these three main entities to retrieval performance tend to overlap; nevertheless, at present, we do not have enough knowledge about this process to say whether, how, and to what extent these entities interact and/or overlap – and how their contributions can be combined, e.g. in a linear fashion or according to some more complex relation.

The above issue is in direct relationship with another long-standing problem in the IR experimentation: the impossibility of testing a single component independently of a complete IRS. [16, p. 12] points out that "if we want to decide between alternative indexing strategies for example, we must use these strategies *as part of a complete information retrieval system*, and examine its overall performance (with each of the alternatives) directly". This means that we have to proceed by changing only one component at time and keeping all the others fixed, in order to identify the impact of that component on retrieval effectiveness;

this also calls for the identification of suitable baselines with respect to which comparisons can be made.

3 The CIRCO Framework

In order to run these grid experiments, we need to set up a framework in which participants can exchange the intermediate output of the components of their systems and create a run by using the output of the components of other participants.

For example, if the expertise of participant A is in building stemmers and decompounders while participant B's expertise is in developing probabilistic IR models, we would like to make it possible for participant A to apply his stemmer to a document collection, pass the output to participant B, who tests his probabilistic IR model, thus obtaining a final run which represents the test of participant A' stemmer + participant B probabilistic IR model.

To this end, the objective of the *Coordinated Information Retrieval Components Orchestration (CIRCO)* framework [9] is to allow for a *distributed, loosely-coupled, and asynchronous* experimental evaluation of *Information Retrieval (IR)* systems where:

- *distributed* highlights that different stakeholders can take part to the experimentation each one providing one or more components of the whole IR system to be evaluated;
- *loosely-coupled* points out that minimal integration among the different components is required to carry out the experimentation;
- *asynchronous* underlines that no synchronization among the different components is required to carry out the experimentation.

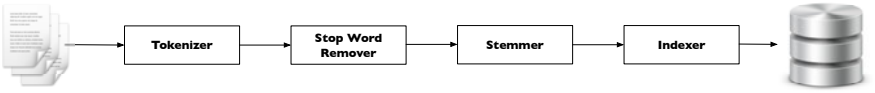
The CIRCO framework allows different research groups and industrial parties, each one with their own areas of expertise, to take part in the creation of *collaborative experiments*. This is a radical departure from today's IR evaluation practice where each stakeholder has to develop (or integrate components to build) an entire IR system to be able to run a single experiment.

The base idea – and assumption – behind CIRCO to streamline the architecture of an IR system and represent it as a *pipeline* of components chained together. The processing proceeds by passing the results of the computations of a component as input to the next component in the pipeline without branches, i.e. no alternative paths are allowed in the chain.

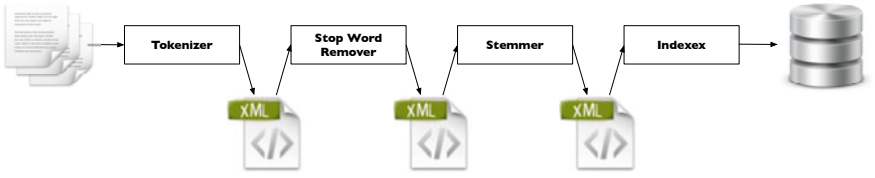
To get an intuitive idea of the overall approach adopted in CIRCO, consider the example pipeline shown in Figure 3(a).

The example IR system is constituted by the following components:

- *tokenizer*: breaks the input documents into a sequence of tokens;
- *stop word remover*: removes stop words from the sequence of tokens;
- *stemmer*: stems the tokens;
- *indexer*: weights the tokens and stores them and the related information in an index.



(a) An example pipeline for an IR system.



(b) An example of CIRCO pipeline for an IR system.

Fig. 3. Example of CIRCO approach to distributed, loosely-coupled, and asynchronous experimentation

Instead of directly feeding the next component as usually happens in an IR system, CIRCO operates by requiring each component to input and output from/to *eXtensible Markup Language (XML)* [25] files in a well-defined format, as shown in Figure 3(b).

These XML files can then be exchanged among the different stakeholders that are involved in the evaluation. In this way, we can meet the requirements stated above by allowing for an experimentation that is:

- *distributed* since different stakeholders can take part in the same experiment, each one providing his own component(s);
- *loosely-coupled* since the different components do not need to be integrated into a whole and running IR system but only need to communicate by means of a well-defined XML format;
- *asynchronous* since the different components do not need to operate all at the same time or immediately after the previous one but can exchange and process the XML files at different rates.

In order to allow this way of conducting experiments, the CIRCO framework consists of:

- *CIRCO Schema*: an XML Schema [23,24] model which precisely defines the format of the XML files exchanged among stakeholders' components;
- *CIRCO Java*[2]: an implementation of CIRCO based on the Java[3] programming language to facilitate its adoption and portability.

[2] The documentation is available at the following address: `http://ims.dei.unipd.it/software/circo/apidoc/`
The source code and the binary code are available at the following address: `http://ims.dei.unipd.it/software/circo/jar/`

[3] `http://java.sun.com/`

The choice of using an XML-based exchange format is due to the fact that the main other possibility, i.e. to develop a common *Application Program Interface (API)* IR systems have to comply with, presents some issues:

- the experimentation would not be *loosely-coupled*, since all the IR systems would have to be coded with respect to the same API;
- much more complicated solutions would be required for allowing the *distributed* and *asynchronous* running of the experiments, since you would need some kind of middleware for process orchestration and message delivery;
- multiple versions of the API in different languages should be provided to take into account the different technologies used to develop IR system;
- the integration with legacy code could be problematic and require a lot of effort;
- overall, stakeholders would be distracted from their main objective, which is running an experiment and evaluating a system.

4 Track Setup

The Grid@CLEF tracks offers a traditional ad-hoc task – see, for example, [1,13] – which makes use of experimental collections developed according to the Cranfield paradigm [5]. This first year task focuses on monolingual retrieval, i.e. querying topics against documents in the same language of the topics, in five European languages: Dutch, English, French, German, and Italian.

The selected languages allow participants to test both romance and germanic languages, as well as languages with word compounding issues. These languages have been extensively studied in the *MultiLingual Information Access (MLIA)* field and, therefore, it will be possible to compare and assess the outcomes of the first year experiments with respect to the existing literature.

This first year track has a twofold goal:

1. to prepare participants' systems to work according to CIRCO framework;
2. to conduct as many experiments as possible, i.e. to put as many dots as possible on the grid.

4.1 Test Collections

Grid@CLEF 2009 used the test collection originally developed for the CLEF 2001 and 2002 campaigns [2,3].

The Documents. Table 1 reports the document collections which have been used for each of the languages offered for the track.

Topics. Topics are structured statements representing information needs. Each topic typically consists of three parts: a brief "title" statement; a one-sentence "description"; a more complex "narrative" specifying the relevance assessment

Table 1. Document collections

Language	Collection	Documents	Size (approx.)
Dutch	NRC Handelsblad 1994/95	84,121	291 Mbyte
	Algemeen Dagblad 1994/95	106,484	235 Mbyte
		190,605	**526 Mbyte**
English	Los Angeles Times 1994	**113,005**	**420 Mbyte**
French	Le Monde 1994	44,013	154 Mbyte
	French SDA 1994	43,178	82 Mbyte
		87,191	**236 Mbyte**
German	Frankfurter Rundschau 1994	139,715	319 Mbyte
	Der Spiegel 1994/95	13,979	61 Mbyte
	German SDA 1994	71,677	140 Mbyte
		225,371	**520 Mbyte**
Italian	La Stampa 1994	58,051	189 Mbyte
	Italian SDA 1994	50,527	81 Mbyte
		108,578	**270 Mbyte**

criteria. Topics are prepared in xml format and uniquely identified by means of a *Digital Object Identifier (DOI)*[4].

In Grid@CLEF 2009, we used 84 out of 100 topics in the set 10.2452/41-AH–10.2452/140-AH originally developed for CLEF 2001 and 2002 since they have relevant documents in all the collections of Table 1.

Figure 4 provides an example of the used topics for all the five languages.

Relevance Assessment. The same relevance assessment developed for CLEF 2001 and 2002 have been used; for further information see [2,3].

4.2 Result Calculation

Evaluation campaigns such as TREC and CLEF are based on the belief that the effectiveness of IRSs can be objectively evaluated by an analysis of a representative set of sample search results. For this, effectiveness measures are calculated based on the results submitted by the participants and the relevance assessments. Popular measures usually adopted for exercises of this type are Recall and Precision. Details on how they are calculated for CLEF are given in [4]. We used `trec_eval`[5] 8.0 to compute the performance measures.

The individual results for all official Grid@CLEF experiments in CLEF 2009 are given in the Appendices of the CLEF 2009 Working Notes [7].

[4] `http://www.doi.org/`
[5] `http://trec.nist.gov/trec_eval`

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<topic>
    <identifier>10.2452/125-AH</identifier>

    <title lang="nl">Gemeenschapplijke Europese munt.</title>
    <title lang="en">European single currency</title>
    <title lang="fr">La monnaie unique européenne</title>
    <title lang="de">Europäische Einheitswährung</title>
    <title lang="it">La moneta unica europea</title>

    <description lang="nl">
        Wat is het geplande tijdschema voor de invoering van de gemeenschapplijke Europese
munt?
    </description>
    <description lang="en">
        What is the schedule predicted for the European single currency?
    </description>
    <description lang="fr">
        Quelles sont les prévisions pour la mise en place de la monnaie unique européenne?
    </description>
    <description lang="de">
        Wie sieht der Zeitplan für die Einführung einer europäischen Einheitswährung aus?
    </description>
    <description lang="it">
        Qual è il calendario previsto per la moneta unica europea?
    </description>

    <narrative lang="nl">
        De veronderstellingen van politieke en economische persoonlijkheden wat betreft het
tijdschema waarbinnen men zal komen tot de invoering van een gemeenschapplijke munt voor de
Europese Unie zijn van belang.
    </narrative>
    <narrative lang="en">
        Speculations by politicians and business figures about a calendar for achieving a
common currency in the EU are relevant.
    </narrative>
    <narrative lang="fr">
        Les débats animés par des personnalités du monde politique et économique sur le
calendrier prévisionnel pour la mise en &#156;uvre de la monnaie unique dans l'Union
Européenne sont pertinents.
    </narrative>
    <narrative lang="de">
        Spekulationen von Vertretern aus Politik und Wirtschaft über einen Zeitplan zur
Einführung einer gemeinsamen europäischen Währung sind relevant.
    </narrative>
    <narrative lang="it">
        Sono rilevanti le previsioni, da parte di personaggi politici e dell'economia, sul
calendario delle scadenze per arrivare a una moneta unica europea.
    </narrative>
</topic>
```

Fig. 4. Example of topic `http://direct.dei.unipd.it/10.2452/125-AH`

5 Track Outcomes

5.1 Participants and Experiments

As shown in Table 2, a total of 2 groups from 2 different countries submitted official results for one or more of the Grid@CLEF 2009 tasks.

Participants were required to submit at least one title+description ("TD") run per task in order to increase comparability between experiments: all the 18 submitted runs used this combination of topic fields. A breakdown into the separate tasks is shown in Table 3.

The participation in this first year was especially challenging because of the need of modifying existing systems to implement the CIRCO framework. Moreover, it has been challenging also from the computational point of view since, for each component in a IR pipeline, CIRCO could produce XML files that are 50-60 times the size of the original collection; this greatly increased the indexing time and the time needed to submit runs and deliver the corresponding XML files.

Table 2. Grid@CLEF 2009 participants

Participant	Institution	Country
chemnitz	Chemnitz University of Technology	Germany
cheshire	U.C.Berkeley	United States

Table 3. Breakdown of experiments into tasks and topic languages

Task	# Participants	# Runs
Monolingual Dutch	0	0
Monolingual English	2	6
Monolingual French	2	6
Monolingual German	2	6
Monolingual Italian	0	0
Total		**18**

5.2 Results

Table 4 shows the top runs for each target collection, ordered by mean average precision. The table reports: the short name of the participating group; the mean average precision achieved by the experiment; the DOI of the experiment; and the performance difference between the first and the last participant.

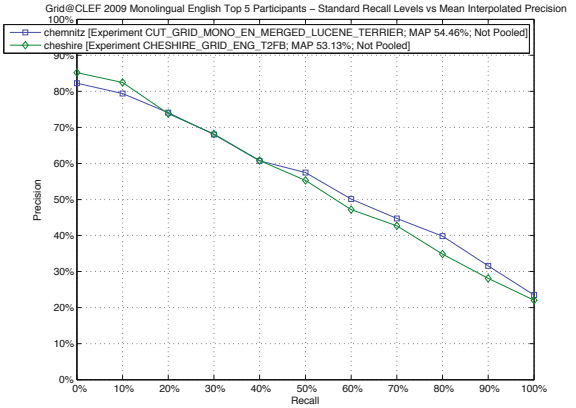
Figure 5 compares the performances of the top participants of the Grid@CLEF monolingual tasks.

6 Approaches and Discussion

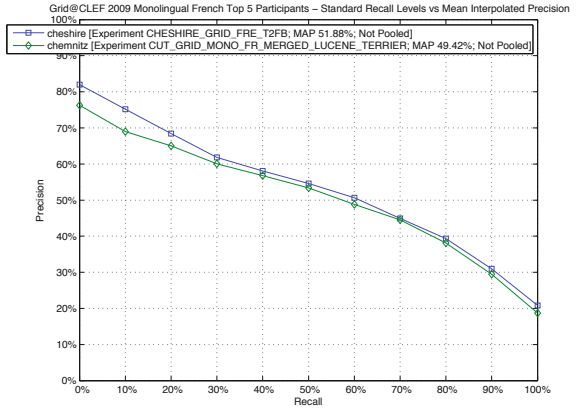
Chemnitz [8] approached the participation in Grid@CLEF into the wider context of the creation of an archive of audiovisual media which can be jointly

Table 4. Best entries for the Grid@CLEF tasks

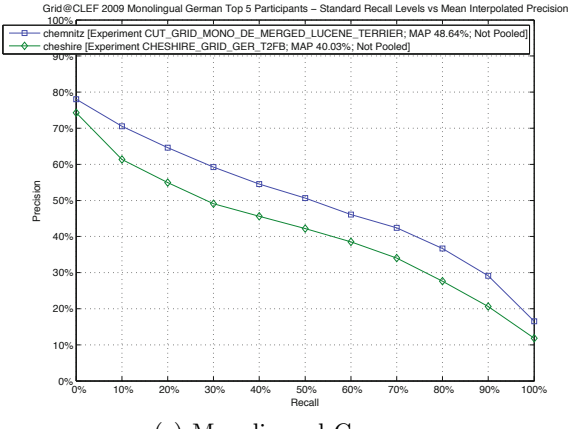
Track	Rank	Participant	Experiment DOI	MAP
English	1st	chemnitz	10.2415/GRIDCLEF-MONO-EN-CLEF2009.CHEMNITZ.CUT_GRID_MONO_EN_MERGED_LUCENE_TERRIER	54.45%
	2nd	chesire	10.2415/GRIDCLEF-MONO-EN-CLEF2009.CHESHIRE.CHESHIRE_GRID_ENG_T2FB	53.13%
	Difference			2.48%
French	1st	chesire	10.2415/GRIDCLEF-MONO-FR-CLEF2009.CHESHIRE.CHESHIRE_GRID_FRE_T2FB	51.88%
	2nd	chemnitz	10.2415/GRIDCLEF-MONO-FR-CLEF2009.CHEMNITZ.CUT_GRID_MONO_FR_MERGED_LUCENE_TERRIER	49.42%
	Difference			4.97%
German	1st	chemnitz	10.2415/GRIDCLEF-MONO-DE-CLEF2009.CHEMNITZ.CUT_GRID_MONO_DE_MERGED_LUCENE_TERRIER	48.64%
	2nd	chesire	10.2415/GRIDCLEF-MONO-DE-CLEF2009.CHESHIRE.CHESHIRE_GRID_GER_T2FB	40.02%
	Difference			21.53%



(a) Monolingual English



(b) Monolingual French



(c) Monolingual German

Fig. 5. Recall-precision graph for Grid@CLEF tasks

used by German TV stations, stores both raw material as well as produced and broadcasted material and needs to be described as comprehensively as possible in order to be easily searchable. In this context, they have developed the Xtrieval system, which aims to be flexible and easily configurable in order to be adjusted to different corpora, multimedia search tasks, and annotation kinds. Chemnitz tested both the vector space model [20,19], as implemented by Lucene[6] and BM25 [17,18], as implemented by Terrier[7], in combination with Snowball[8] and Savoy's [21] stemmers. They found out that the impact of retrieval techniques are highly depending on the corpus and quite unpredictable and that, even if over they years they have learned how to guess reasonable configurations for their system in order to get good results, there is still the need of "strong rules which let us predict the retrieval quality ... [and] enable us to automatically configure a retrieval engine in accordance to the corpus". This was for them motivation to participate in Grid@CLEF 2009, which represented a first attempt that will allow them to go also in this direction.

Chesire [14] participated in Grid@CLEF with their Chesire II system based on logistic regression [6] and their interest was in understanding what happens when you try to separate the processing elements of IR systems and look at their intermediate output, taking this as an opportunity to re-analyse and improve their system, and, possibly, finding a way to incorporate into Chesire II components of other IR systems for subtasks in which they currently cannot do or cannot do effectively, such as decompounding German words. They also found that "the same algorithms and processing systems can have radically different performance on different collections and query sets". Finally, the participation in Grid@CLEF actually allowed Cheshire to improve their system and to point out some suggestions for the next Grid@CLEF, concerning the support for the creation of multiple indexes according to the structure of a document and specific indexing tasks related to the geographic information retrieval, such as geographic names extraction and geo-referencing.

Acknowledgements

The authors would like to warmly thank the members of the Grid@CLEF Advisory Committee – Martin Braschler, Chris Buckley, Fredric Gey, Kalervo Järvelin, Noriko Kando, Craig Macdonald, Prasenjit Majumder, Paul McNamee, Teruko Mitamura, Mandar Mitra, Stephen Robertson, and Jacques Savoy – for the useful discussions and suggestions.

The work reported has been partially supported by the TrebleCLEF Coordination Action, within FP7 of the European Commission, Theme ICT-1-4-1 Digital Libraries and Technology Enhanced Learning (Contract 215231).

[6] http://lucene.apache.org/
[7] http://ir.dcs.gla.ac.uk/terrier/index.html
[8] http://snowball.tartarus.org/

References

1. Agirre, E., Di Nunzio, G.M., Ferro, N., Mandl, T., Peters, C.: CLEF 2008: Ad Hoc Track Overview. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 15–37. Springer, Heidelberg (2009)
2. Braschler, M.: CLEF 2001 – Overview of Results. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.) CLEF 2001. LNCS, vol. 2406, pp. 9–26. Springer, Heidelberg (2002)
3. Braschler, M.: CLEF 2002 – Overview of Results. In: Peters, C., Braschler, M., Gonzalo, J. (eds.) CLEF 2002. LNCS, vol. 2785, pp. 9–27. Springer, Heidelberg (2003)
4. Braschler, M., Peters, C.: CLEF 2003 Methodology and Metrics. In: Peters, C., Gonzalo, J., Braschler, M., Kluck, M. (eds.) CLEF 2003. LNCS, vol. 3237, pp. 7–20. Springer, Heidelberg (2004)
5. Cleverdon, C.W.: The Cranfield Tests on Index Languages Devices. In: Spärck Jones, K., Willett, P. (eds.) Readings in Information Retrieval, pp. 47–60. Morgan Kaufmann Publisher, Inc., San Francisco (1997)
6. Cooper, W.S., Gey, F.C., Dabney, D.P.: Probabilistic Retrieval Based on Staged Logistic Regression. In: Belkin, N.J., Ingwersen, P., Mark Pejtersen, A., Fox, E.A. (eds.) Proc. 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1992), pp. 198–210. ACM Press, New York (1992)
7. Di Nunzio, G.M., Ferro, N.: Appendix D: Results of the Grid@CLEF Track. In: Borri, F., Nardi, A., Peters, C. (eds.) Working Notes for the CLEF 2009 Workshop (2009) (published online)
8. Eibl, M., Kürsten, J.: Putting It All Together: The Xtrieval Framework at Grid@CLEF. In: Peters et al. [15] (2009)
9. Ferro, N.: Specification of the CIRCO Framework, Version 0.10. Technical Report IMS. 2009. CIRCO.0.10, Department of Information Engineering, University of Padua, Italy (2009)
10. Ferro, N., Harman, D.: Dealing with MultiLingual Information Access: Grid Experiments at TrebleCLEF. In: Agosti, M., Esposito, F., Thanos, C. (eds.) Post-proceedings of the Fourth Italian Research Conference on Digital Library Systems (IRCDL 2008), pp. 29–32. ISTI-CNR at Gruppo ALI, Pisa (2008)
11. Ferro, N., Peters, C.: From CLEF to TrebleCLEF: the Evolution of the Cross-Language Evaluation Forum. In: Kando, N., Sugimoto, M. (eds.) Proc. 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, pp. 577–593. National Institute of Informatics, Tokyo (2008)
12. Ferro, N., Peters, C.: CLEF Ad-hoc: A Perspective on the Evolution of the Cross-Language Evaluation Forum. In: Agosti, F. Esposito, C. Thanos, editors, Post-proceedings of the Fifth Italian Research Conference on Digital Library Systems (IRCDL 2009), pp. 72–79. DELOS Association and Department of Information Engineering of the University of Padua (2009)
13. Ferro, N., Peters, C.: CLEF, Ad Hoc Track Overview: TEL & Persian Tasks. In: Peters et al. [15] (2009)
14. Larson, R.R.: Decomposing Text Processing for Retrieval: Cheshire tries GRID@CLEF. In: Peters et al. [15]

15. Peters, C., Di Nunzio, G.M., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.): Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments – Tenth Workshop of the Cross–Language Evaluation Forum (CLEF 2009). Revised Selected Papers. LNCS. Springer, Heidelberg (2010)
16. Robertson, S.E.: The methodology of information retrieval experiment. In: Spärck Jones, K. (ed.) Information Retrieval Experiment, pp. 9–31. Butterworths, London (1981)
17. Robertson, S.E., Spärck Jones, K.: Relevance Weighting of Search Terms. Journal of the American Society for Information Science (JASIS) 27(3), 129–146 (1976)
18. Robertson, S.E., Walker, S., Beaulieu, M.: Experimentation as a way of life: Okapi at TREC. Information Processing & Management 36(1), 95–108 (2000)
19. Salton, G., Buckley, C.: Term-weighting Approaches in Automatic Text Retrieval. Information Processing & Management 24(5), 513–523 (1988)
20. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM (CACM) 18(11), 613–620 (1975)
21. Savoy, J.: A Stemming Procedure and Stopword List for General French Corpora. Journal of the American Society for Information Science (JASIS) 50(10), 944–952 (1999)
22. Savoy, J.: Report on CLEF-2001 Experiments: Effective Combined Query-Translation Approach. In: Peters, C., Braschler, M., Gonzalo, J., Kluck, M. (eds.) CLEF 2001. LNCS, vol. 2406, pp. 27–43. Springer, Heidelberg (2002)
23. W3C. XML Schema Part 1: Structures – W3C Recommendation (October 28, 2004), http://www.w3.org/TR/xmlschema-1/
24. W3C. XML Schema Part 2: Datatypes – W3C Recommendation (October 28, 2004), http://www.w3.org/TR/xmlschema-2/
25. W3C. Extensible Markup Language (XML) 1.0 (Fifth Edition) – W3C Recommendation (November 26, 2008), http://www.w3.org/TR/xml/

Decomposing Text Processing for Retrieval: Cheshire Tries GRID@CLEF

Ray R. Larson

School of Information
University of California, Berkeley, USA
ray@sims.berkeley.edu

Abstract. This short paper describes Berkeley's participation in the GRID@CLEF task. The GRID@CLEF task is intended to capture in XML form the intermediate results of the text processing phases of the indexing process used by IR systems. Our approach was to create a new instrumented version of the indexing program used with the Cheshire II system. Thanks to an extension by the organizers, we were able to submit runs derived from our system.

The system used for this task is a modified version of the Cheshire II IR system, to which output files for the different intermediate streams have been added. The additions, like the original system were written in C. Developing this system required creating parallel modules for several elements of the Cheshire II indexing programs. The current version handles the simplest processing cases, and currently ignores the many specialized indexing modes in the system (such as geographic name extraction and georeferencing).

1 Introduction

The Berkeley Cheshire group decided to participate in GRID@CLEF for two primary reasons. The first was that the task goal of separating the processing elements of IR systems and looking at the their intermediate output was interesting. The second was more concerned with a detailed reanalysis of our existing processing system and the hope of finding new and better ways to do some of the things that we have developed over the past decade. Since one goal of the GRID@CLEF task is for systems to be able to both export and import intermediate processing streams and eventually to share them, we also hope to be able to use others' streams as inputs for subtasks in which we currently cannot do or cannot do effectively (such as decompounding German words).

The system that we used for GRID@CLEF is a modified version of the Cheshire II IR system, which we have used for all of our participation in various CLEF tracks over the past several years. The modifications made to the system (for this year) primarily concerned the pre-processing and "normalization" of text. In the current implemention of the GRID-enabled system the indexing program is primarily affected. Essentially the indexing program retains all of the functionality that it previously had, but now it will generate output XML files for

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 566–569, 2010.

the different intermediate streams during the text processing and normalization process. These additions, like the original system were written in C. Developing the modified system required creating parallel modules for several elements of the Cheshire II indexing program. Those modules needed to pass along data from a higher level in the call tree down to the low-level code where functions were called to output tokens, stems, etc. to the appropriate files. There are a myriad of alternative parsing approaches, etc. controlled by Cheshire II configuration files, and in this first-cut version for GRID@CLEF only a very few of the most basic ones are supported. Because the system developed over time to support a variety of speciallized index modes and features (such as extracting and georeferencing place names from texts to permit such things as geographic searching though proximity, and extracting dates and times in such a way that they can be searched by time ranges, etc. instead of treating dates as character strings). For the current implementation of we deal only with text extraction and indexing, and do not even attempt to deal with separate indexes for different parts of the documents.

2 Information Retrieval Approach

For retrieval in the GRID@CLEF track we used the same algorithms that we used in other CLEF participation (including for GikiCLEF and Adhoc-TEL this year), without change. In fact, the basic processing captured by the output files submitted for this track has been fairly standard for our participation across all tracks in CLEF. For retrieval, we used the inverted file and vector file indexes created during the indexing process using the same Logistic Regression-based ranking algorithm that we have used elsewhere[1].

3 Text Processing Result Submissions

For GRID@CLEF in addition to the conventional retrieval runs (described in the next section), we submitted four intermediate streams from the indexing process. These were:

Basic tokens – in Cheshire II parsing into tokens takes place once an XML subtree of a document required for a particular index specified in a configuration file is located. To keep things as simple as possible in this version, the XML sub-tree is the entire document (e.g., the <doc> tag and all of its descendents). Tokenization first eliminates all XML tags in the subtree (replacing them with blanks) and then uses the "strtok" C string library function to include any sequence of alphanumeric characters divided at white space or punctuation (with the exception of hyphens and periods, which are retained at this point). Hyphens are treated specially and double extracted, once as the hyphenated word and then as separate words with the hyphen(s) removed. (At least that is what it SHOULD be doing – in checking results for this stage I found that only the first word of a hyphenated word was getting

extracted. This is now being corrected). Sequences of letters and periods are assumed to be initialisms (like U.S.A.) and are left in the basic token stream.

Lowercase normalization – The default normalization (which can be turned of by the configuration files) is to change all characters to lowercase. This step also removes any trailing period from tokens (so U.S.A. becomes u.s.a).

Stopword removal – Each index can have an index-specific stoplist and any words matching those in the stoplist are thrown out and don't go on to any later stages.

Stemming – For each collection the configuration file specified use of particular stemmers including the Snowball stemmer for various languages and an extended version of the Porter stemmer. The Snowball stemming system has been integrated into the Cheshire II system and any of its stemmers can be invoked via different configuration file options.

Finally the remaining stemmed tokens are accumulated along with their document frequency information and stored in a temporary file. In subsequent stages the information for all of the documents is sorted, merged and an inverted file created from the tokens and their document frequency information.

In retrieval, the same stages are performed on the tokens derived from the topics or queries before matching takes place.

The XML files produced for each of these streams ranged in size from 18.5Gb for raw token files to 4.5Gb after stemming, depending on the test collection and the position in processing.

4 Retrieval Results

Although our retrieval runs were submitted quite late, the organizers kindly allowed them to go through the same evaluation as the officially submitted runs. We submitted only one monolingual run for each of English, French, and German.

The indexes and vector files created during the later stages of the indexing process (and not yet captured by the GRID@CLEF output streams) were used to provide the matching used in the logistic regression algorithm described above. Overall, the retrieval results look fairly good (although there was only one other participant to compare with) with comparable results in all languages (except German, where I suspect the other group is using decompounding).

Figure 1 shows the precision-recall graph for all of our submitted runs. The MAP of our German run was the lowest at 0.4003, with a MAPs of 0.5313 and 0.5188 for English and French, respectively. Interestingly, the identical algorithm and processing (without capturing the intermediate outputs) was used in our Adhoc-TEL participation this year, with much worse performance in terms of average precision when compared to even the same group also participating in this task, which shows that the same algorithms and processing systems can have radically different performance on different collections and query sets.

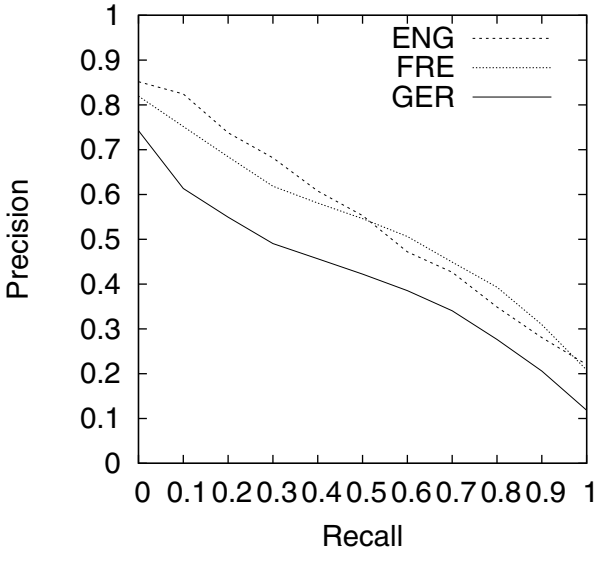


Fig. 1. Berkeley Monolingual Runs, English, French, and German

5 Conclusions

One of the goals in our participation in GRID@CLEF was to identify problems and issues with our text processing and normalization stages. In that we have been quite successful, having identified one definite bug and a number of areas for re-design and enhancement. The next phase would be to enable the system to take any of the intermediate streams produced by different participants as input. This is a much more difficult problem, since much further work and analysis is needed. Since, for example the Cheshire system can create separate indexes based on different parts of an XML or SGML record, the streams would also need to carry this kind of information along with them. In addition, some of our indexing methods perform the text processing in different sequences (for example, geographic name extraction uses capitalization as one way of identifying proper nouns that might be place names, and the output of the georeferencing process is a set of geographic coordinates instead of a text name).

Overall this has been a very interesting and useful track and provided several improvements to our system that will carry over to other tasks as well.

References

1. Larson, R.R.: Cheshire at geoclef 2007: Retesting text retrieval baselines. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 811–814. Springer, Heidelberg (2008)

Putting It All Together: The Xtrieval Framework at Grid@CLEF 2009

Jens Kürsten and Maximilian Eibl

Chemnitz Technical University, Straße der Nationen 62, 09107 Chemnitz
{kuersten,eibl}@informatik.tu-chemnitz.de

Abstract. The Xtrieval framework, built at the Chemnitz University of Technology, aims at analyzing the impact of different retrieval models and methods on retrieval effectiveness. For the Grid@CLEF task 2009, the CIRCO framework was integrated into the Xtrieval framework. 15 runs were performed 15 runs in the three languages German, English, and French. For each language two different stemmers and two different retrieval models were used. One run was a fusion run combining the results of the four other experiments. Whereas the different runs demonstrated that the impact of the used retrieval technologies is highly depending on the corpus, the merged approach produced the best results in each language.

1 Introduction

It is a well known fact that retrieval methods perform in a different way on different corpora. Nevertheless, the exact way the methods perform is still enigmatic. In order to understand the impact of the single retrieval methods in accordance to the used corpus the media computer science group at Chemnitz University of Technology started to design and implement a highly flexible retrieval framework in 2005. The aim has been to gain in-depth insight into the effects of the different retrieval techniques in order to apply them to a real world problem: an archive for audiovisual media.

In 2006 the group started participating at CLEF achieving results from acceptable to very good at many different tasks. By now we can claim we got some kind of gut instinct how to configure our system in order to produce good results. But we did not get much closer to gain knowledge about the impact of retrieval techniques based on hard facts.

Until now, the Xtrieval framework has become flexible enough to start an extensive cross evaluation of retrieval methods in relation to corpus types. The Grid@CLEF task heads into a similar direction with two exceptions: First, it focuses on multilingual retrieval. Though the Xtrieval framework can deal with different languages multilinguality has not been is not the main focus of our evaluation. Second, the Grid tasks define four different retrieval steps in order to do the evaluation. This approach is necessary in order to manage the different participating groups. But it is somewhat restrictive since the retrieval process has to be partitioned in exactly the four steps.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 570–577, 2010.

This contribution provides a short description of the Xtrieval framework and its accompanying annotation framework AMOPA. A detailed description of the background and the frameworks is provided by the working notes [1]. The final section discusses our results at the Grid@CLEF task in 2009. Some general thoughts and an outlook to future work concludes the paper.

2 The Xtrieval and AMOPA Frameworks

In 2005 we started conceptualizing and implementing a flexible framework for information retrieval purposes. It is a general finding in information retrieval, that the performance of retrieval systems highly depends on hardly generalizable aspects like for example corpora: Retrieval methods that perform well with one corpus do not necessarily work at all when applied to another corpus. After all that is the reason for installing different tracks in evaluation campaigns like CLEF[1] and TREC[2].

The general idea was to create a framework which is highly flexible and adjustable concerning information retrieval technologies. The framework needed to provide interfaces (API) to combine different state-of-the-art text retrieval techniques on the one hand and to evaluate and integrate new methods for multimedia retrieval on the other hand. An in-depth description of the framework design is given in [2].

The framework, named Xtrieval, implements a Java-based object-orientated API specification providing interfaces to all methods necessary for all possible designs of retrieval systems. By this, the framework is able to exchange, evaluate, and combine different components of other retrieval systems. In a first implementation Apache Lucene[3] was integrated but by now also Terrier[4] and Lemur[5] are included in practice. The framework supports not only the integration of these and other toolkits but also allows combining their retrieval results on the fly.

Thus, the framework provides a realm of possible configurations. In order to conveniently adjust the system to different corpora we created a Graphical User Interface (GUI). This GUI provides a general configuration interface that supports the user in setting all parameter driven classes. Thus, all parameters of each class can be changed during runtime without any changes in the source code of the project. A second interface incorporates methods for calculating and visualizing recall-precision graphs. Additional functions to load and save relevance assessments in popular formats (e.g. TREC) are provided as well.

The GUI can be used to configure the three main components: indexing, retrieval and evaluation. A general programming interface is able to convert every structured data collection into an internal representation which is then used for the application of transformation and tokenization procedures like for example different stemming algorithms. The pre-processed data is than passed forward to a programming interface which allows connecting indexing libraries like Lucene. In order to integrate the full amount of metadata of audiovisual data we created the framework AMOPA.

[1] http://www.clef-campaign.org/
[2] http://trec.nist.gov
[3] http://lucene.apache.org/
[4] http://ir.dcs.gla.ac.uk/terrier/
[5] http://www.lemurproject.org/

Probably the most important interface of the Xtrieval framework allows the flexible use of retrieval algorithms. Queries are pre-processed according to the needs of different toolkits. It is also possible to combine searches in different indexes and to fuse these results into one result set by for example Sum-RSV, Product-RSV, and Z-Score.

Finally the evaluation component is capable to store and reload experiments and their complete parameter sets. This enables us to repeat experiments at a later date. It provides several measures to compare retrieval output to assessments. Additionally, it is possible to load and store relevance assessments in the TREC format.

For practical reasons (video analysis tools are written in C, Xtrieval in Java) we built for the automated annotation tasks a separate framework called AMOPA-Automated MOving Picture Annotator. AMOPA uses the FFMPEG[6] library to read video stream and perform first low level methods. Access for Java code to the C library FFMPEG is provided by the library FFMPEG-Java, which is part of the Streambaby[7] project. The actual analysis is performed by AMOPA and organized in process chains. This concept allows us to exchange and reorder processes very easily. A detailed description of AMOPA is given in [3].

3 Grid Retrieval in 2009

In 2009 we participated the 4th time at CLEF. [1] gives a summary of our experiences with different CLEF tasks (Image, Video, QA, Ad-Hoc, Wikimedia) and provides short insight into the experiences of other participating groups. Detailed information of our results is provided by [4], [5], [6], [7], [8], [9], [10], [11], [12], and [13]. Interestingly, our system performed quite different over the years and the tasks. Performance seems to be highly depending on the underlying corpus.

The Xtrieval framework was used to prepare and run our text retrieval experiments for the Grid Experiments Pilot Track. The core retrieval functionality is provided by Apache Lucene, the Lemur toolkit, and the Terrier framework. This allowed us to choose from a wide range of state of the art retrieval models for all kinds of text retrieval experiments. Our main goal in this first Grid experiment was to provide strong baseline experiments, which could be used as reference for evaluation of sophisticated new retrieval approaches.

In order to participate at the Grid@CLEF track the CIRCO framework [14] had to be integrated into Xtrieval. Since one of the main design concepts of the Xtrieval framework was flexibility towards enhancements only a small number of classes had to be rewritten: two classes that are used to process the token streams during indexing and another class that writes the processed token stream in the index format of the used retrieval core. Since the integration of the Lemur and Terrier retrieval toolkits into Xtrieval had been done lately we did not have the time to test and debug the integration. Thus, we decided to adapt the Lucene indexing class only.

[6] http://ffmpeg.org/
[7] http://code.google.com/p/streambaby/

Ten collections in five European languages, namely Dutch, English, French, German and Italian were provided for the Grid Experiment Pilot Track. For our participation we chose to run experiments on the English, French and German collections, which included six text collections in total. Table 1 shows the used collections and the provided fields which were taken for indexing. Table 2 shows some indexing statistics.

Table 1. Fields used for indexing

Collection	Indexed Fields
DE: Spiegel 1994/5	LEAD, TEXT, TITLE
DE: Frankfurter Rundschau 1994	TEXT, TITLE
DE: German SDA 1994	KW, LD, NO, ST, TB, TI, TX
EN: LA Times 1994	BYLINE, HEADLINE, TEXT
FR: Le Monde 1994	CHA1, LEAD1, PEOPLE, SUBJECTS, TEXT, TIO1
FR: French SDA 194	KW, LD, NO, ST, TB, TI, TX

Table 2. Index statistics and CIRCO output per language

Lang	Stemmer	# Docs	# Terms	# Distinct Terms	# Chunk Files	Compr. File Size (MB)
DE	Snowball	225,371	28,711,385	3,365,446	225	15,695
DE	N-gram Decomp	225,371	63,118,598	840,410	225	19,924
EN	Snowball	113,005	20,210,424	685,141	114	14,293
EN	Krovetz	113,005	20,701,670	704,424	114	14,293
FR	Snowball	87,191	12,938,610	1,130,517	88	7,329
FR	Savoy [16]	87,191	13,262,848	1,239,705	88	7,323

We performed 15 runs, five for each language German, English, and French. For each language we used two different stemmers and two different retrieval models. One run one was a fusion run combining the results of the four other experiments. Table 4 provides the general configuration of each experiment as well as the retrieval performance in terms of mean average precision (MAP) and geometric mean average precision (GMAP). Please note the French run cut_fr_3. This run was corrupted while submitting. We did a separate evaluation for this run: cut_fr_3* is not part of the official statistics but shows the correct results.

All in all, merging models and stemmers brings the best results for all three languages. Comparing the models and stemmers leads to the following conclusions:

- German: BM25 performs better than VSM. N-gram performs better than Snowball.
- English: The results in English are vice versa: VSM performs (slightly) better than BM25. Snowball performs (slightly) better than Krovetz.

- French: Here the results are even more confusing: VSM performs (especially in conjunction with Savoy) better than BM25. In conjunction with VSM Snowball performs better but in conjunction with BM25 Savoy is superior.

Table 3. Results overview

Lang	ID	Core	Model	Stemmer	# QE docs / tokens	MAP	GMAP
DE	cut_de_1	Lucene	VSM	Snowball	10 / 50	.4196	.2023
DE	cut_de_2	Terrier	BM25	Snowball	10 / 50	.4355	.2191
DE	cut_de_3	Lucene	VSM	N-gram	10 / 250	.4267	.2384
DE	cut_de_4	Terrier	BM25	N-gram	10 / 250	.4678	.2682
DE	**cut_en_5**	**both**	**both**	**both**	**10 / 50 & 250**	**.4864**	**.3235**
EN	cut_en_1	Lucene	VSM	Snowball	10 / 20	.5067	.3952
EN	cut_en_2	Terrier	BM25	Snowball	10 / 20	.4926	.3314
EN	cut_en_3	Lucene	VSM	Krovetz	10 / 20	.4937	.3762
EN	cut_en_4	Terrier	BM25	Krovetz	10 / 20	.4859	.3325
EN	**cut_en_5**	**both**	**both**	**both**	**10 / 20**	**.5446**	**.4153**
FR	cut_fr_3	Lucene	VSM	Snowball	10 / 20	.0025	.0000
FR	cut_fr_3*	Lucene	VSM	Snowball	10 / 20	.4483	.3060
FR	cut_fr_1	Terrier	BM25	Snowball	10 / 20	.4538	.3141
FR	cut_fr_5	Lucene	VSM	Savoy [16]	10 / 20	.4434	.2894
FR	cut_fr_2	Terrier	BM25	Savoy [16]	10 / 20	.4795	.3382
FR	**cut_fr_4**	**both**	**both**	**both**	**10 / 20**	**.4942**	**.3673**

Thus, some results demonstrate a better performance for VSM, some results show superiority of BM25. The results for the stemmers are similarly unpredictable. But it seems that this uncertainty can be overcome by data fusion: As table 4 demonstrates, for each language the merging of the retrieval models produced the best results. In our framework, merging is done by the z-score operator [16]. The results for the merged experiments are shown in figure 1.

These results confirm our findings described in section 2: the impact of retrieval techniques are highly depending on the corpus and quite unpredictable. All in all, while participating at CLEF we developed a decent gut instinct in configuring the retrieval system to produce good and very good retrieval results. But in fact the configuring task is still at bit like stumbling in the dark. The exact effects of retrieval mechanisms remain enigmatic. We still do not have strong rules which let us predict the retrieval quality. And so we never know whether or not there is a better configuration we did not predict. Having such rules would enable us to automatically configure a retrieval engine in accordance to the corpus.

It is our belief, that far more experiments are needed before we get even close to such rules. The Grid@CLEF track is exactly the platform the community needs to answer this question. Unfortunately, this year we were the only participants. Thus, our results lack some expressiveness. We will certainly take part again next year and hope to be not the only ones then.

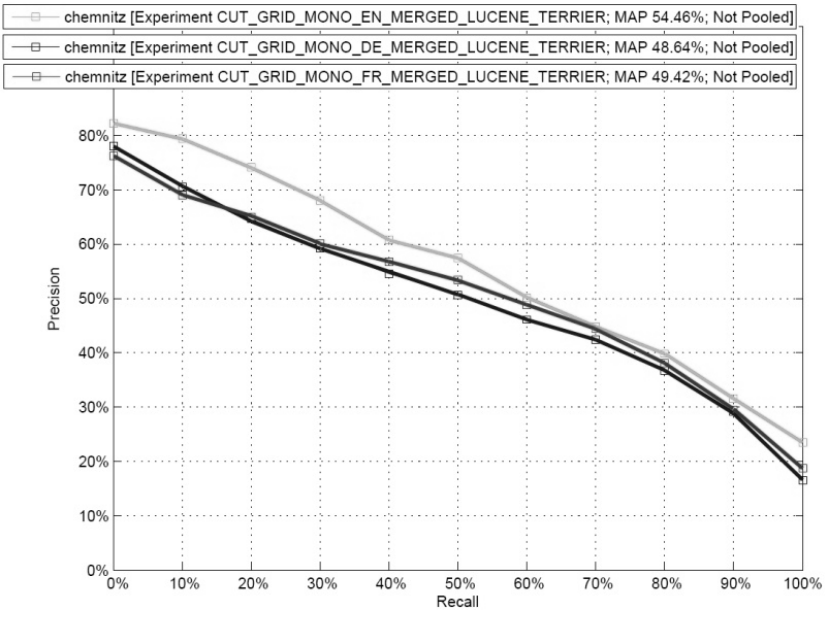


Fig. 1. Results for the merged experiments

Acknowledgements

We would like to thank Jaques Savoy and his co-workers for providing numerous resources for language processing. Also, we would like to thank Giorgio M. di Nunzio and Nicola Ferro for developing and operating the DIRECT system.

This work was partially accomplished in conjunction with the project sachsMedia, which is funded by the Unternehmen Region-program[8] of the German Federal Ministry of Education and Research[9].

References

All Web references were retrieved on August 24th, 2009.

1. Eibl, M., Kürsten, J.: The Importance of being Grid: Chemnitz University of Technology at Grid@CLEF. In: Working Notes for the CLEF 2009, Workshop (2009), http://www.clef-campaign.org/2009/working_notes/kuersten-videoclef-paperCLEF2009.pdf
2. Wilhelm, T.: Entwurf und Implementierung eines Frameworks zur Analyse und Evaluation von Verfahren im Information Retrieval. Chemnitz University of Technology, Faculty of Computer Science, Diploma Thesis (2007), http://archiv.tu-chemnitz.de/pub/2008/0117/data/da.pdf

[8] http://www.unternehmen-region.de
[9] http://www.bmbf.de/en/

3. Ritter, M.: Visualisierung von Prozessketten zur Shot Detection. In: Workshop Audiovisuelle Medien WAM 2009: Archivierung, pp.135–150 (2009), http://archiv.tu-chemnitz.de/pub/2009/0095/data/wam09_monarch.pdf
4. Kürsten, J., Eibl, M.: Monolingual Retrieval Experiments with a Domain-Specic Document Corpus at the Chemnitz Technical University. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 178–185. Springer, Heidelberg (2007)
5. Wilhelm, T., Eibl, M.: ImageCLEF, Experiments at the Chemnitz Technical University. In: Working Notes for the CLEF 2006, Workshop (2006), http://www.clef-campaign.org/2006/working_notes/workingnotes2006/wilhelmCLEF2006l.pdf
6. Kürsten, J., Eibl, M.: Domain-Spefic Cross Language Retrieval: Comparing and Merging Structured and Unstructured Indices. In: Working Notes for the CLEF 2007, Workshop (2007), http://www.clef-campaign.org/2007/working_notes/kuerstenCLEF2007.pdf
7. Wilhelm, T., Kürsten, J., Eibl, M.: Experiments for the ImageCLEF 2007, Photographic Retrieval Task. In: Working Notes for the CLEF 2007, Workshop (2007), http://www.clef-campaign.org/2007/working_notes/wilhelmCLEF2007.pdf
8. Kürsten, J., Wilhelm, T., Eibl, M.: CLEF 2008, Ad-Hoc Track: On-line Processing Experiments with Xtrieval. In: Working Notes for the CLEF 2008, Workshop (2008), http://www.clef-campaign.org/2008/working_notes/kuersten-ah-paperCLEF2008.pdf
9. Kürsten, J., Wilhelm, T., Eibl, M.: The Xtrieval Framework at CLEF 2008: Domain-Specic Track. In: Working Notes for the CLEF 2008, Workshop (2008), http://www.clef-campaign.org/2008/working_notes/kuersten-ds-paperCLEF2008.pdf
10. Kürsten, J., Richter, D., Eibl, M.: VideoCLEF 2008: ASR Classification based on Wikipedia Categories. In: Working Notes for the CLEF 2008, Workshop (2008), http://www.clef-campaign.org/2008/working_notes/kuersten-videoclef-paperCLEF2008.pdf
11. Wilhelm, T., Kürsten, J., Eibl, M.: The Xtrieval Framework at CLEF 2008: Image CLEF photographic retrieval task. In: Working Notes for the CLEF 2008, Workshop (2008), http://www.clef-campaign.org/2008/working_notes/wilhelm-imageclefphoto-paperCLEF2008.pdf
12. Wilhelm, T., Kürsten, J., Eibl, M.: The Xtrieval Framework at CLEF 2008: Image CLEF Wikipedia MM task. In: Working Notes for the CLEF 2008, Workshop (2008), http://www.clef-campaign.org/2008/working_notes/wilhelm-imageclefwiki-paperCLEF2008.pdf
13. Kürsten, J., Kundisch, H., Eibl, M.: QA Extension for Xtrieval: Contribution to the QAst track. In: Working Notes for the CLEF 2008, Workshop (2008), http://www.clef-campaign.org/2008/working_notes/kuersten-qast-paperCLEF2008.pdf
14. Ounis, I., Lioma, C., Macdonald, C., Plachouras, V.: Research directions in terrier: a search engine for advanced retrieval on the web. Novatica/UPGRADE Special Issue on Next Generation Web Search, 49–56 (2007)

15. Ferro, N., Harman, D.: Dealing with multilingual information access: Grid experiments at trebleclef. In: Post-proceedings of the Fourth Italian Research Conference on Digital Library Systems (IRCDL 2008), pp. 29–32 (2008)
16. Savoy, J.: Data Fusion for Effective European Monolingual Information Retrieval. In: Working Notes for the CLEF 2004, Workshop, Bath, UK, September 15-17 (2004), http://www.clef-campaign.org/2004/working_notes/WorkingNotes2004/22.pdf

Overview and Results of Morpho Challenge 2009

Mikko Kurimo[1], Sami Virpioja[1], Ville T. Turunen[1],
Graeme W. Blackwood[2], and William Byrne[2]

[1] Adaptive Informatics Research Centre, Aalto University,
P.O. Box 15400, FIN-00076 Aalto, Finland
http://www.cis.hut.fi/morphochallenge2009/
[2] Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, U.K.

Abstract. The goal of Morpho Challenge 2009 was to evaluate unsupervised algorithms that provide morpheme analyses for words in different languages and in various practical applications. Morpheme analysis is particularly useful in speech recognition, information retrieval and machine translation for morphologically rich languages where the amount of different word forms is very large. The evaluations consisted of: 1. a comparison to grammatical morphemes, 2. using morphemes instead of words in information retrieval tasks, and 3. combining morpheme and word based systems in statistical machine translation tasks. The evaluation languages were: Finnish, Turkish, German, English and Arabic. This paper describes the tasks, evaluation methods, and obtained results. The Morpho Challenge was part of the EU Network of Excellence PASCAL Challenge Program and organized in collaboration with CLEF.

1 Introduction

Unsupervised morpheme analysis is one of the important but unsolved tasks in computational linguistics and its applications, such as speech recognition (ASR) [1,2], information retrieval (IR) [3,4] and statistical machine translation (SMT) [5,6]. The morphemes are useful, because the lexical modeling using words is particularly problematic for the morphologically rich languages, such as Finnish, Turkish and Arabic. In those languages the number of different word forms is very large because of various inflections, prefixes, suffixes and compound words.

It is possible to construct rule based tools that perform morphological analysis quite well, but of the large number of languages in the world, only few have such tools available. This is because the work of human experts to generate the rules or annotate the morpheme analysis of words and texts is expensive. Thus, learning to perform the analysis based on unannotated text collections is an important goal. Even for those languages that already have existing analysis tools, the statistical machine learning methods still propose interesting and competitive alternatives.

The scientific objectives of the Morpho Challenge are: to learn about the word construction in natural languages, to advance machine learning methodology,

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 578–597, 2010.

and to discover approaches that are suitable for many languages. In Morpho Challenge 2009, the participants first developed unsupervised algorithms and submitted their analyses of the word lists in different languages provided by the organizers. Then various evaluations were carried out using the proposed morpheme analysis to find out how they performed in different tasks. In 2009 Challenge the evaluations consisted of both a comparison to grammatical morphemes (*Competition 1*), IR (*Competition 2*) and SMT (*Competition 3*) tasks. The IR experiments contain CLEF tasks, where the all the words in the queries and text corpus were replaced by their morpheme analyses. In SMT experiments identical translation systems using the same data are first trained using morpheme analysis and words, and then combined for the best performance. The SMT tasks were first time introduced this year for Morpho Challenge and are based on recent work of the organizers in morpheme based machine translation [6,7].

2 Participants and Their Submissions

By the submission deadline in 8th August, 2009, ten research groups had submitted algorithms, which were then evaluated by the organizers. The authors and the names of their algorithms are listed in Table 1. The total number of tasks that the algorithms were able to participate in was 11: six in Competition 1, three in Competition 2, and two in Competition 3. The submissions for the different tasks are presented in Table 2. The final number of algorithms per task varied from 6 to 15.

Table 1. The participants and the names of their algorithms. The short acronyms of max 8 characters are used in the result tables throughout the paper.

Authors, Affiliations:	Algorithm name	[Acronym]
D. Bernhard, TU Darmstadt, D:	MorphoNet	[MorphNet]
B. Can & S. Manandhar, U. York, UK:	1 [CanMan1], 2	[CanMan2]
B. Golénia et al., U. Bristol, UK:	UNGRADE	[Ungrade]
J-F. Lavallée & P. Langlais, U. Montreal, CA:	RALI-ANA	[Rali-ana],
	RALI-COF	[Rali-cof]
C. Lignos et al., U. Penn. & Arizona, USA:	-	[Lignos]
C. Monson et al., Oregon Health & Sc. U., USA:	ParaMor Mimic	[P-Mimic],
ParaMor-Morfessor Mimic [PM-Mimic], ParaMor-Morfessor Union		[PM-Union]
S. Spiegler et al., U. Bristol, UK:	PROMODES	[Prom-1],
PROMODES 2 [Prom-2], PROMODES committee		[Prom-com]
T. Tchoukalov et al., U. Stanford & OHSU, USA:	MetaMorph	[MetaMorf]
S. Virpioja & O. Kohonen, Helsinki U. Tech., FI:	Allomorfessor	[Allomorf]

Statistics of the output of the submitted algorithms are briefly presented in Table 3 for English. The corresponding data for each of the languages is presented in [8]. The average amount of analyses per word is shown in the column

Table 2. The submitted analyses for Arabic (non-vowelized and vowelized), English, Finnish, German and Turkish. C2 means the additional English, Finnish and German word lists for Competition 2. C3 means the Finnish and German word lists for Competition 3.

Algorithm	ARA-NV	ARA-V	ENG	FIN	GER	TUR	C2	C3
MorphNet	X	X	X	X	X	X	X	X
CanMan1	-	-	X	-	X	X	-	-
CanMan2	-	-	-	-	X	X	-	-
Ungrade	X	X	X	X	X	X	-	-
Rali-ana	X	X	X	X	X	X	-	-
Rali-cof	X	X	X	X	X	X	-	-
Lignos	-	-	X	-	X	-	-	-
P-Mimic	X	X	X	X	X	X	X	X
PM-Mimic	X	X	X	X	X	X	X	X
PM-Union	X	X	X	X	X	X	X	X
Prom-1	X	X	X	X	X	X	-	-
Prom-2	X	X	X	X	X	X	-	-
Prom-com	X	X	X	X	X	X	-	-
MetaMorf	X	X	X	X	X	X	-	X
Allomorf	X	X	X	X	X	X	X	X
Total	12	12	14	12	15	14	5	6

"#analyses". It is interesting that in contrary to previous years, now all algorithms ended up mostly suggesting only one analysis per word. From the column "#morphs" we see the average amount of morphemes per analysis, which reflects the level of details the algorithm provides. The total amount of morpheme types in the lexicon is given in the column "#types".

As baseline results for unsupervised morpheme analysis, the organizers provided morpheme analysis by a publicly available unsupervised algorithm called "Morfessor Categories-MAP" (or "Morfessor CatMAP, CatMAP " for short) developed at Helsinki University of Technology [9]. Analysis by the original Morfessor [10,11] (or here "Morfessor Baseline, MorfBase"), which provides only a surface-level segmentation, was also provided for reference. Additionally, the reference results were provided for "letters", where the words are simply split into letters, and "Gold Standard", which is a linguistic gold standard morpheme analysis.

3 Competition 1 – Comparison to Linguistic Morphemes

3.1 Task and Data

The task was to return the given list of words in each language with the morpheme analysis added after each word. It was required that the morpheme analyses should be obtained by an unsupervised learning algorithm that would preferably be as

Table 3. Statistics and example morpheme analyses in **English**. #analyses is the average amount of analyses per word (separated by a comma), #morphs the average amount of morphemes per analysis (separated by a space), and #types the total amount of morpheme types in the lexicon.

Algorithm	#analyses	#morphs	#types	example analysis
MorphNet	1	1.75	211439	vulnerabilty _ies
CanMan	1	2.09	150097	vulner abilities
Ungrade	1	3.87	123634	vulnerabilities
Rali-ana	1	2.10	166826	vulner abiliti es
Rali-cof	1	1.91	145733	vulnerability ies
Lignos	1	1.74	198546	VULNERABILITY +(ies)
P-Mimic	1	3.03	188716	vulner +a +bilit +ie +s
PM-Mimic	1	2.96	166310	vulner +a +bilit +ies
PM-Union	1	2.87	120148	vulner a +bilit +ies
Prom-1	1	3.28	107111	vul nerabilitie s
Prom-2	1	3.63	47456	v ul nera b ili ties
Prom-com	1	3.63	47456	v ul nera b ili ties
MetaMorf	1	1.58	241013	vulnerabiliti es
Allomorf	1	2.59	23741	vulnerability ies
MorfBase	1	2.31	40293	vulner abilities
CatMAP	1	2.12	132038	vulner abilities
letters	1	9.10	28	v u l n e r a b i l i t i e s
Gold Standard	1.06	2.49	18855	vulnerable_A ity_s +PL

language independent as possible. In each language, the participants were pointed to a training corpus in which all the words occur (in a sentence), so that the algorithms may also utilize information about the word context. The tasks were the same as in the Morpho Challenge 2008 last year.

The training corpora were the same as in the Morpho Challenge 2008, except for Arabic: 3 million sentences for English, Finnish and German, and 1 million sentences for Turkish in plain unannotated text files that were all downloadable from the Wortschatz collection[1] at the University of Leipzig (Germany). The corpora were specially preprocessed for the Morpho Challenge (tokenized, lowercased, some conversion of character encodings).

For Arabic, we tried this year a very different data set, the Quran, which is smaller (only 78K words), but has also a vowelized version (as well as the unvowelized one) [12]. The corresponding full text data was also available. In Arabic, the participants could try to analyze the vowelized words or the unvowelized, or both. They were evaluated separately against the vowelized and the unvowelized gold standard analysis, respectively. For all Arabic data, the Arabic writing script were provided as well as the Roman script (Buckwalter transliteration `http://www.qamus.org/transliteration.htm`.). However, only morpheme analysis submitted in Roman script, was evaluated.

[1] `http://corpora.informatik.uni-leipzig.de/`

The exact syntax of the word lists and the required output lists with the suggested morpheme analyses have been explained in [13]. As the learning is unsupervised, the returned morpheme labels may be arbitrary: e.g., "foot", "morpheme42" or "+PL". The order in which the morpheme labels appear after the word forms does not matter. Several interpretations for the same word can also be supplied, and it was left to the participants to decide whether they would be useful in the task, or not.

In Competition 1 the proposed unsupervised morpheme analyses were compared to the correct grammatical morpheme analyses called here the linguistic gold standard. The gold standard morpheme analyses were prepared in exactly the same format as the result file the participants were asked to submit, alternative analyses separated by commas. For the other languages except Arabic, the gold standard reference analyses were the same as in the Morpho Challenge 2007 [13]. For Arabic the gold standard has in each line; the word, the root, the pattern and then the morphological and part-of-speech analysis. See Table 4 for examples.

Table 4. Examples of gold standard morpheme analyses

Language	Examples	
English	baby-sitters	baby_N sit_V er_s +PL
	indoctrinated	in_p doctrine_N ate_s +PAST
	vulnerabilities	vulnerable_A ity_s +PL
Finnish	linuxiin	linux_N +ILL
	makaronia	makaroni_N +PTV
German	eu-jsenmaissa	eu jäsen_N maa_N +PL +INE
	choreographische	choreographie_N isch +ADJ-e
	zurueckzubehalten	zurueck_B zu be halt_V +INF
	durchliefen	durch_P lauf_V +PAST +13PL
Turkish	kontrole	kontrol +DAT
	popUlerliGini	popUler +DER_lHg +POS2S +ACC,
		popUler +DER_lHg +POS3 +ACC3
	CukurlarIyla	Cukur +PL +POS3 +REL,
		Cukur +POS3S +REL
Arabic Vowelized	AdoEuwAniy	dEw faEala dEuw +Verb +Imperative +2P
		+Pl +Pron +Dependent +1P
Arabic Unvowelized	AdEwny	dEw fEl dEw +Verb +Imperative +2P
		+Pl +Pron +Dependent +1P

3.2 Evaluation

The evaluation of Competition 1 in Morpho Challenge 2009 was similar as in Morpho Challenges 2007 and 2008, but a few changes were made to the evaluation measure: small bugs related to the handling of alternative analyses were fixed from the scripts and points were now measured as one per word, not one per word pair.

Because the morpheme analysis candidates are achieved by unsupervised learning, the morpheme labels can be arbitrary and different from the ones designed by linguists. The basis of the evaluation is, thus, to compare whether any two word forms that contain the same morpheme according to the participants' algorithm also has a morpheme in common according to the gold standard and vice versa. The proportion of morpheme sharing word pairs in the participant's sample that really has a morpheme in common according to the gold standard is called the *precision*. Correspondingly, the proportion of morpheme sharing word pairs in the gold standard sample that also exist in the participant's submission is called the *recall*.

In practise, the precision was calculated as follows: A number of word forms were randomly sampled from the result file provided by the participants; for each morpheme in these words, another word containing the same morpheme was chosen from the result file by random (if such a word existed). We thus obtained a number of word pairs such that in each pair at least one morpheme is shared between the words in the pair. These pairs were compared to the gold standard; a point was given if the word pair had at least the same number of common morphemes according to the gold standard as they had in the proposed analysis. If the gold standard had common morphemes, but less than proposed, fractions of points were given. In the case of alternative analyses in the gold standard, the best matching alternative was used. The maximum number of points for one sampled word was normalized to one. The total number of points was then divided by the total number of sampled words. The sample size in different languages varied depending on the size of the word lists and gold standard: 200,000 (Finnish), 50,000 (Turkish), 50,000 (German), 10,000 (English), and 5,000 (Arabic) word pairs.

For words that had several alternative analyses, as well as for word pairs that have more than one morpheme in common, normalization of the points was carried out. In short, an equal weight is given for each alternative analysis, as well as each word pair in an analysis. E.g., if a word has three alternative analyses, the first analysis has four morphemes, and the first word pair in that analysis has two morphemes in common, each of the two common morphemes will amount to $1/3 * 1/4 * 1/2 = 1/24$ of the one point available for that word.

The recall was calculated analogously to precision: A number of word forms were randomly sampled from the gold standard file; for each morpheme in these words, another word containing the same morpheme was chosen from the gold standard by random (if such a word existed). The word pairs were then compared to the analyses provided by the participants; a full point was given for each sampled word pair that had at least as many morphemes in common also in the analyses proposed by the participants' algorithm. Again, points per word was normalized to one and the total number of points was divided by the total number of words.

The *F-measure*, which is the harmonic mean of precision and recall, was selected as the final evaluation measure:

$$\text{F-measure} = 1/(1/\text{Precision} + 1/\text{Recall})\,. \tag{1}$$

Table 5. The morpheme analyses compared to the gold standard in **non-vowelized and vowelized Arabic** (Competition 1). The numbers are in %.

Method	Non-vowelized Arabic Precision	Recall	F-measure	Method	Vowelized Arabic Precision	Recall	F-measure
letters	70.48	53.51	60.83	letters	50.56	84.08	63.15
Prom-2	76.96	37.02	50.00	Prom-2	63.00	59.07	60.97
Prom-com	77.06	36.96	49.96	Prom-com	68.32	47.97	56.36
Prom-1	81.10	20.57	32.82	Ungrade	72.15	43.61	54.36
Ungrade	83.48	15.95	26.78	Prom-1	74.85	35.00	47.70
Allomorf	91.62	6.59	12.30	MorfBase	86.87	4.90	9.28
MorfBase	91.77	6.44	12.03	PM-Union	91.61	4.41	8.42
MorphNet	90.49	4.95	9.39	Allomorf	88.28	4.37	8.33
PM-Union	93.72	4.81	9.14	PM-Mimic	93.62	3.23	6.24
PM-Mimic	93.76	4.55	8.67	MorphNet	92.52	2.91	5.65
Rali-ana	92.40	4.40	8.41	MetaMorf	88.78	2.89	5.59
MetaMorf	95.05	2.72	5.29	Rali-ana	91.30	2.83	5.49
P-Mimic	91.29	2.56	4.97	P-Mimic	91.36	1.85	3.63
Rali-cof	94.56	2.13	4.18	Rali-cof	95.09	1.50	2.95

3.3 Results

The results of the Competition 1 are presented in Tables 5 and 6. In three languages, Turkish, Finnish and German, the algorithms with clearly highest F-measures were "ParaMor-Morfessor Mimic" and "Union". In English, however, "Allomorfessor" was better and also the algorithm by Lignos et al. was quite close. In Arabic, the results turned out quite surprising, because most algorithms gave rather low recall and F-measure and nobody was able to beat the simple "letters" reference. "Promodes" and "Ungrade" methods scored clearly better than the rest of the participants in Arabic.

The tables contain also results of the best algorithms from Morpho Challenges 2008 [14] [PM-2008], [ParaMor] and 2007 [13] [Bernhard2], [Bordag5a]. From Morpho Challenge 2008, the best method "Paramor + Morfessor 2008" [PM-2008] would have also scored highest in 2009. However, this was a combination of two separate algorithms, ParaMor and Morfessor, where the two different analyses were just given as alternative analyses for each word. As the evaluation procedure selects the best matching analysis, this boosts up the recall, while obtaining precision that is about the average of the two algorithms. By combining this year's top algorithms in a similar manner, it would be easy to get even higher scores. However, exploiting this property of the evaluation measure is not a very interesting approach.

Excluding "Paramor + Morfessor 2008", this year's best scores for the English, Finnish, German and Turkish tasks are higher than the best scores in 2008. However, Bernhard's second method from 2007 [Bernhard2] holds still the highest score for English, Finnish and German. The best result for the Turkish task has improved yearly.

Table 6. The morpheme analyses compared to the gold standard in %. The results below the line are by the winners of the previous Morpho Challenges.

	English				German		
Method	Precision	Recall	F-measure	Method	Precision	Recall	F-measure
Allomorf	68.98	56.82	62.31	PM-Union	52.53	60.27	56.14
MorfBase	74.93	49.81	59.84	PM-Mimic	51.07	57.79	54.22
PM-Union	55.68	62.33	58.82	CatMAP	71.08	38.92	50.30
Lignos	83.49	45.00	58.48	P-Mimic	50.81	47.68	49.20
P-Mimic	53.13	59.01	55.91	CanMan2	57.67	42.67	49.05
MorphNet	65.08	47.82	55.13	Rali-cof	67.53	34.38	45.57
PM-Mimic	54.80	60.17	57.36	Prom-2	36.11	50.52	42.12
Rali-cof	68.32	46.45	55.30	Allomorf	77.78	28.83	42.07
CanMan1	58.52	44.82	50.76	MorphNet	67.41	30.19	41.71
CatMAP	84.75	35.97	50.50	Prom-1	49.88	33.95	40.40
Prom-1	36.20	64.81	46.46	Prom-com	48.48	34.61	40.39
Rali-ana	64.61	33.48	44.10	MorfBase	81.70	22.98	35.87
Prom-2	32.24	61.10	42.21	Lignos	78.90	21.35	33.61
Prom-com	32.24	61.10	42.21	Ungrade	39.02	29.25	33.44
MetaMorf	68.41	27.55	39.29	MetaMorf	39.59	19.81	26.40
Ungrade	28.29	51.74	36.58	CanMan1	73.16	15.27	25.27
letters	3.82	99.88	7.35	Rali-ana	61.39	15.34	24.55
				letters	2.79	99.92	5.43
PM-2008	69.59	65.57	67.52	PM-2008	64.06	61.52	62.76
ParaMor	63.32	51.96	57.08	ParaMor	70.73	38.82	50.13
Bernhard2	67.42	65.11	66.24	Bernhard2	54.02	60.77	57.20

	Finnish				Turkish		
Method	Precision	Recall	F-measure	Method	Precision	Recall	F-measure
PM-Union	47.89	50.98	49.39	PM-Mimic	48.07	60.39	53.53
PM-Mimic	51.75	45.42	48.38	PM-Union	47.25	60.01	52.88
CatMAP	79.01	31.08	44.61	P-Mimic	49.54	54.77	52.02
Prom-com	41.20	48.22	44.44	Rali-cof	48.43	44.54	46.40
P-Mimic	47.15	40.50	43.57	CatMAP	79.38	31.88	45.49
Prom-2	33.51	61.32	43.34	Prom-2	35.36	58.70	44.14
Prom-1	35.86	51.41	42.25	Prom-1	32.22	66.42	43.39
Rali-cof	74.76	26.20	38.81	MorphNet	61.75	30.90	41.19
Ungrade	40.78	33.02	36.49	CanMan2	41.39	38.13	39.70
MorphNet	63.35	22.62	33.34	Prom-com	55.30	28.35	37.48
Allomorf	86.51	19.96	32.44	Ungrade	46.67	30.16	36.64
MorfBase	89.41	15.73	26.75	MetaMorf	39.14	29.45	33.61
MetaMorf	37.17	15.15	21.53	Allomorf	85.89	19.53	31.82
Rali-ana	60.06	10.33	17.63	MorfBase	89.68	17.78	29.67
letters	5.17	99.89	9.83	Rali-ana	69.52	12.85	21.69
				letters	8.66	99.13	15.93
				CanMan1	73.03	8.89	15.86
PM-2008	65.21	50.43	56.87	PM-2008	66.78	57.97	62.07
ParaMor	49.97	37.64	42.93	ParaMor	57.35	45.75	50.90
Bernhard2	63.92	44.48	52.45	Bordag5a	81.06	23.51	36.45

4 Competition 2 – Information Retrieval

In Competition 2, the morpheme analyses were compared by using them in IR tasks with three languages: English, German and Finnish. The tasks and corpora were the same as in 2007 [4] and 2008 [15]. In the evaluation, words occurring in the corpus and in the queries were replaced by the morpheme segmentations submitted by the participants. Additionally, there was an option to access the test corpus and evaluate the IR performance using the morpheme analysis of word forms in their full text context.

Morpheme analysis is important in a text retrieval task because the user will want to retrieve all documents irrespective of which word forms are used in the query and in the text. Of the tested languages, Finnish is the most complex morphologically and is expected to gain most from a successful analysis. Compound words are typical of German while English is morphologically the simplest.

4.1 Task and Data

In a text retrieval task, the user formulates their information need to a query and the system has to return all documents from the collection that satisfy the user's infomation need. To evaluate the performance of a retrieval system, a collection of documents, a number of test queries and a set of human relevance assessments are needed.

In Competition 2, the participants' only task was to provide segmentations for the given word lists. The word lists were extracted from the test corpora and queries. In addition, the words in the Competition 1 word lists were added to the Competition 2 lists. Optionally, the participants could also register to the Cross-Language Evaluation Forum (CLEF)[2] and use the full text corpora for preparing the morpheme analysis. The IR experiments were performed by the Morpho Challenge organizers by using the submitted word lists to replace the words both in the documents and in the queries by their proposed analyses.

The corpora, queries and relevance assessments were provided by CLEF and contained news paper articles as follows:

- In Finnish: 55K documents from short articles in Aamulehti 1994-95, 50 test queries on specific news topics and 23K binary relevance assessments (CLEF 2004)
- In English: 170K documents from short articles in Los Angeles Times 1994 and Glasgow Herald 1995, 50 test queries on specific news topics and 20K binary relevance assessments (CLEF 2005).
- In German: 300K documents from short articles in Frankfurter Rundschau 1994, Der Spiegel 1994-95 and SDA German 1994-95, 60 test queries with 23K binary relevance assessments (CLEF 2003).

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

4.2 Reference Methods

The participants' submissions were compared against a number of relevant reference methods which were the same as used in Morpho Challenge 2008 [15]. Like the participants' methods, Morfessor baseline [MorfBase] [10,11] and Morfessor Categories-MAP [CatMAP] [9] are unsupervised algorithms. Also evaluated were a commercial word normalization tool [TWOL] and the rule-based grammatical morpheme analyses [gram] based on the linguistic gold standards [16]. These methods have the benefit of language specific linguistic knowledge embedded in them. Because some words may have several alternative interpretations two versions of these references cases were given: either all alternatives were used (e.g. [TWOL-all]) or only the first one (e.g. [TWOL-1]). Traditional language-dependent stemming approaches based on the Snowball libstemmer library [StemEng], [StemGer] and [StemFin] as well as using the words without any processing were also tested [dummy].

In each task the best algorithm in 2008, i.e. the one that provided the highest average precision ([PM-2008] and [McNamee4]) can be used as a reference, too, because the IR tasks in 2009 were identical to 2008.

4.3 Evaluation

English, German and Finnish IR tasks were used to evaluate the submitted morpheme analyses. Unfortunately, neither Turkish or Arabic IR test corpora were available for the organizers. The experiments were performed by replacing the words in the corpus and the queries by the submitted morpheme analyses. Thus, the retrieval was based on morphemes as index terms. If a segmentation for a word was not provided, it was left unsegmented and used as a separate morpheme. The queries were formed by using the title and description ("TD") fields from the topic descriptions.

The IR experiments were performed using the freely available LEMUR toolkit[3] version 4.4. The popular Okapi BM25 ranking function was used. In the 2007 challenge [4], it was noted that the performance of Okapi BM25 suffers greatly if the corpus contains morphemes that are very common. The unsupervised morpheme segmentation algorithms tend to introduce such morphemes when they e.g. separate suffixes. To overcome this problem, a method for automatically generating a stoplist was introduced. Any term that has a collection frequency higher than 75000 (Finnish) or 150000 (German and English) is added to the stoplist and thus excluded from the corpus. Even though the method is quite simplistic, it generates reasonable sized stoplists (about 50-200 terms) and is robust with respect to the cutoff parameter. With a stoplist, Okapi BM25 clearly outperformed TFIDF ranking and thus the approach has been adopted for later evaluations as well. The evaluation criterion for the IR performance is the Mean Average Precision (MAP) that was calculated using the `trec_eval` program.

[3] `http://www.lemurproject.org/`

4.4 Results

Three research groups submitted total of five different segmentations for the Competition 2 word lists. In addition, for the 6 groups and 10 algorithms that did not provide segmentations for the Competition 2 word lists, the smaller Competition 1 word list was used. None of the participants used the option to use the full text corpora to provide analyses for words in their context.

Table 7. The obtained mean average precision (MAP) in the IR tasks. Asterisk (*) denotes submissions that did not include segmentations for Competition 2 and were evaluated by using the shorter Competition 1 word list. The results below the line are statistically significantly different from the best result of that language.

Method	English	Method	German	Method	Finnish
StemEng	0.4081	TWOL-1	0.4885	TWOL-1	0.4976
PM-2008	0.3989	TWOL-all	0.4743	McNamee4	0.4918
TWOL-1	0.3957	PM-2008	0.4734	TWOL-all	0.4845
TWOL-all	0.3922	MorfBase	0.4656	PM-Union	0.4713
Lignos	0.3890*	CatMAP	0.4642	Allomorf	0.4601
MorfBase	0.3861	PM-Mimic	0.4490	CatMAP	0.4441
Allomorf	0.3852	PM-Union	0.4478	MorfBase	0.4425
P-Mimic	0.3822	Allomorf	0.4388	gram-1	0.4312
PM-Union	0.3811	CanMan1	0.4006*	PM-Mimic	0.4294
gram-1	0.3734	Rali-cof	0.3965*	StemFin	0.4275
CatMAP	0.3713	CanMan2	0.3952*	gram-all	0.4090
Rali-ana	0.3707*			Prom-2	0.3857*
PM-Mimic	0.3649			P-Mimic	0.3819
MetaMorf	0.3623*				
Rali-cof	0.3616*				
MorphNet	0.3560				
gram-all	0.3542				
dummy	0.3293	StemGer	0.3865	Rali-cof	0.3740*
Ungrade	0.2996*	Lignos	0.3863*	MorphNet	0.3668
CanMan1	0.2940*	P-Mimic	0.3757	Ungrade	0.3636*
Prom-1	0.2917*	MetaMorf	0.3752*	Rali-ana	0.3595*
Prom-2	0.2066*	Prom-com	0.3634*	dummy	0.3519
Prom-com	0.2066*	dummy	0.3509	Prom-com	0.3492*
		Ungrade	0.3496*	Prom-1	0.3392*
		Prom-1	0.3484*	MetaMorf	0.3289*
		gram-1	0.3353		
		Rali-ana	0.3284*		
		MorphNet	0.3167		
		gram-all	0.3014		
		Prom-2	0.2997*		

Table 7 shows the obtained MAP values for the submissions in English, German and Finnish. For English, the best performance was achieved by the algorithm by Lignos et al. even though only the shorter Competition 1 word list was

available for evaluation. "ParaMor-Morfessor Mimic" and "ParaMor-Morfessor Union" by Monson et. al gave the best performance for German and Finnish respectively. Overall, the algorithms by Monson et al., especially "ParaMor-Morfessor Union", gave good performance across all tested languages. Also, "Allomorfessor" by Virpioja & Kohonen was a solid performer in all languages. However, none of the submitted algorithms could beat the winners of last year's competition.

In all languages, the best performance was achieved by one of the reference algorithms. The rule based word normalizer, TWOL, gave best performance in German and Finnish. In the English task, TWOL was only narrowly beaten by the traditional Porter stemmer. For German and Finnish, stemming was not nearly as efficient. Of the other reference methods, "Morfessor Baseline" gave good performance in all languages while the "grammatical" reference based on linguistic analyses did not perform well probably because the gold standards are quite small.

4.5 Statistical Testing

For practical reasons, a limited set of queries (50-60) are used in evaluation of the IR-performance. The obtained results will include variation between queries as well as between methods. Statistical testing was employed to determine what differences in performance between the submissions are greater than expected by pure chance. The methodology we use follows closely the one used in TREC [17] and CLEF [18].

Analysis was performed with Two-way ANOVA using MATLAB Statistics Toolbox. Since ANOVA assumes the samples to be normally distributed, a transformation for the average precision values was made with the arcsin-root function:

$$f(x) = \arcsin(\sqrt{x}). \tag{2}$$

The transformation makes the samples more normally distributed. Statistical significances were examined using MATLAB's `multcompare` function with the Tukey t-test and 0.05 confidence level.

Based on the confidence test results a horizontal line is drawn in Table 7 at the point where all the methods below it are significantly different from the best result, and the "top group" above it are those that have no significant difference to the best result of each language. Further analysis of the confidence test results are in [8]. The confidence intervals are relatively wide and a large proportion of the submissions are in the top group for all languages. It is well known and also noted in the CLEF Ad Hoc track [18] that it is hard to obtain statistically significant differences between retrieval results with only 50 queries.

One interesting comparison is to see if there are significant differences to the "dummy" case where no morphological analysis is performed. For German and Finnish, "ParaMor-Morfessor Union" is the only submission that is significantly better than the dummy method. For English, none of the participants' results can significantly improve over "dummy". Only the Porter stemmer is significantly better according to the test.

4.6 Discussions

The results of the Competition 2 suggest that unsupervised morphological analysis is a viable approach for IR. Some of the unsupervised methods were able to beat the "dummy" baseline and the best were close to the language specific rule-based "TWOL" word normalizer. However, this year's competition did not offer any improvements to previous results.

The fact that segmentations of the full Competition 2 word list was not provided by all participants makes the comparison of IR performance a bit more difficult. The participants that were evaluated using only the Competition 1 word lists had a disadvantage, because then the additional words in the IR task were indexed as such without analysis. In the experiments in Morpho Challenge 2007 [4], the segmentation of the additional words improved performance in the Finnish task for almost all participants. In German and English tasks the improvements were small. However, if the segmentation algorithm is not performing well, leaving some of the words unsegmented only improves the results for that participant.

Most of the methods that performed well in the Competition 2 IR task were also strong in the corresponding linguistic evaluation of Competition 1 and vice versa. The biggest exeptions were in the Finnish task where the "PROMODES committee" algorithm gave reasonably good results in the linguistic evaluation but not in the IR task. The algorithm seems to oversegment words and the suggested morphemes give good results when compared to gold standard analysis but do not seem to work well as index terms. On the other hand, "Allomorfessor" and the "Morfessor Baseline"methods performed well in the IR task but were not at the top in the linguistic evaluation where they suffered from low recall. In general, it seems that precision in the Competition 1 evaluation is a better predictor of IR performance than recall or F-measure.

The statistical testing revealed very few significant differences in the IR performance between participants. This is typical for the task. However, we feel that testing the algorithms in a realistic application gives information about the performance of the algorithms that the linguistic comparison can not offer alone.

The participants were offered a chance to access the IR corpus to use the full text context in the unsupervised morpheme analysis. Although using the context of words seems a natural way to improve the models none of the participants have attempted this. Other future work includes expanding the IR task to new languages like Arabic which pose new kinds of morphological problems.

5 Competition 3 – Statistical Machine Translation

In Competition 3, the morpheme analyses proposed by the participants' algorithm were evaluated in a SMT framework. The translation models were trained to translate from a morphologically complex source language to English. The words of the source language were replaced by their morpheme analyses before training the translation models. The two source languages used in the competition were Finnish and German. Both the input data for the participants'

algorithms and training the SMT system were from the proceedings of the European Parliament. The final SMT systems were evaluated by measuring the similarity of the translation results to a human-made reference translation.

5.1 Task and Data

As a data set, we used Finnish-English and German-English parts of the European Parliament parallel corpus (release v2) [19]. The participants were given a list of word forms extracted from the corpora, and similarly to the Competitions 1 and 2, they were asked to apply their algorithms to the word list, and return the morphological analyses for the words. It was also possible to use the context information of the words by downloading the full corpus. Furthermore, the data sets from Competitions 1 and 2 were allowed to use for training the morpheme analyses. However, they were used by none of the participants.

For training and testing the SMT systems, the Europarl data sets were divided into three subsets: training set for training the models, development set for tuning the model parameters, and test set for evaluating the translations. For the Finnish-English systems, we had 1 180 603 sentences for training, 2 849 for tuning, and 3 000 for testing. For the German-English systems, we had 1 293 626 sentences for training, 2 665 for tuning, and 3 000 for testing.

5.2 Evaluation

In principle, the evaluation is simple: First, we train a translation system that can translate the morphologically analyzed Finnish or German sentence to English. Then, we use it to translate new sentences, and compare the translation results to the reference translations. If the morphological analysis is good, it reduces the sparsity of the data and helps the translation task. If the analysis contains many errors, they should degrade the translation results. However, a SMT system has many components and parameters that can affect the overall results. Here we describe the full evaluation procedure in detail.

As the SMT models and tools are mainly designed for word-based translations, the results obtained for morpheme-based models are rarely better than the word-based baseline models (see, e.g., [6]). Thus, following the approach in [7], we combined the morpheme-based models to a standard word-based model by generating n-best lists of translation hypotheses from both models, and finding the best overall translation with the Minimum Bayes Risk (MBR) decoding.

Training Phrase-Based SMT Systems. The individual models, including the baseline word-to-word model and the morpheme-to-word models based on the participants' methods, were trained with the open source Moses system [20]. Moses translates sequences of tokens, called phrases, at a time. The decoder finds the most probable hypothesis as a sequence of target language tokens, given a sequence of tokens in source language, a language model, a translation model and possible additional models, such as a reordering model for phrases in the hypothesis.

Training a translation model with Moses includes three main steps: (1) alignment of the tokens in the sentence pairs (2) extracting the phrases from the aligned data, and (3) scoring the extracted phrases. As there are more morphemes than words in a sentence, two limitations affect the results: First, the alignment tool cannot align sentences longer than 100 tokens. Second, the phrases have a maximum length, which we set to be 10 for the morpheme-based models.

The weights of the different components (translation model, language model, etc.) are tuned by maximizing the BLEU score [21] for the development set. Finally, we generated n-best list for the development and test data for the MBR combination. At most 200 distinct hypotheses were generated for each sentence; less if the decoder could not find as many.

Minimum Bayes-Risk Decoding for System Combination. Minimum Bayes-Risk (MBR) decoding for machine translation [22] selects the translation hypothesis that has the lowest expected risk given the underlying probabilistic model. For loss function L bounded by maximum loss L_{max}, we choose the hypothesis that maximises the conditional expected gain according to the decision rule

$$\hat{E} = \underset{E' \in \mathcal{E}}{\operatorname{argmax}} \sum_{E \in \mathcal{E}} G(E, E') P(E|F), \tag{3}$$

where $G(E, E') = L_{max} - L(E, E')$ is the gain between reference E and hypothesis E' and $P(E|F)$ is the posterior probability of translation. The search is performed over all hypotheses E' in the evidence space $\mathcal{E}$, typically an n-best list or lattice. An appropriate gain function for machine translation is the sentence-level BLEU score [21]. For efficient application to both n-best lists and lattices, our MBR decoder uses an approximation to the sentence-level BLEU score formulated in terms of n-gram posterior probabilities [23]. The contribution of each n-gram w is a constant θ_w multiplied by the number of times w occurs in E' or zero if it does not occur. The decision rule is then

$$\hat{E} = \underset{E' \in \mathcal{E}}{\operatorname{argmax}} \left\{ \theta_0 |E'| + \sum_{w \in \mathcal{N}} \theta_w \#_w(E') p(w|\mathcal{E}) \right\}, \tag{4}$$

where $p(w|\mathcal{E})$ is the posterior probability of the n-gram w and $\mathcal{N} = \{w_1, \ldots, w_{|\mathcal{N}|}\}$ denotes the set of all n-grams in the evidence space. The posterior probabilities are computed efficiently using the OpenFst toolkit [24].

We used minimum Bayes-risk system combination [25] to combine n-best list evidence spaces generated by multiple MT systems. The posterior probability of n-gram w in the union of two n-best lists $\mathcal{E}_1$ and $\mathcal{E}_2$ is computed as a linear interpolation of the posterior probabilities according to each individual list:

$$p(w|\mathcal{E}_1 \cup \mathcal{E}_2) = \lambda P(w|\mathcal{E}_1) + (1 - \lambda) P(w|\mathcal{E}_2). \tag{5}$$

The parameter λ determines the weight associated with the output of each translation system and was optimized for BLEU score on the development set.

Evaluation of the Translations. For evaluation of the performance of the SMT systems, we applied BLEU scores [21]. BLEU is based on the co-occurrence of n-grams: It counts how many n-grams (for $n = 1, \ldots, 4$) the proposed translation has in common with the reference translations and calculates a score based on this. Although BLEU is a very simplistic method, it usually corresponds well to human evaluations if the compared systems are similar enough. In our case they should be very similar, as the only varying factor is the morphological analysis. In addition to the MBR combinations, we calculated the BLEU scores for all the individual systems.

5.3 Results

Six methods from four groups were included in Competition 3. In addition, Morfessor Baseline [MorfBase], Morfessor Categories-MAP [CatMAP] and grammatical morphemes [gram-1] were tested as reference methods. We calculated the BLEU scores both for the individual systems, including a word-based system [words], and for MBR combination with the word-based system. The results are in Table 8.

Between the results from the MBR combinations, only some of the differences are statistically significant. The significances were inspected with paired t-test on ten subsets of the test data. In the Finnish to English task, Morfessor Baseline, Allomorfessor, Morfessor CatMAP and MetaMorph are all significantly better than the rest of the algorithms. Between them, the difference between Allomorfessor and the both Morfessor algorithms is not significant, but Allomorfessor and Morfessor Baseline are significantly better than MetaMorph. The differences between the results of the last four algorithms (MorphoNet and ParaMor:s) are not statistically significant. Neither they are significantly better than the word-based system alone.

In the German to English task, only the results of Morfessor Baseline and Allomorfessor have significant differences to the rest of the systems. Morfessor Baseline is significantly better than any of the others expect Allomorfessor and ParaMor Mimic. Allomorfessor is significantly better than the others expect Morfessor Baseline, ParaMor Mimic, ParaMor-Morfessor Mimic and Morfessor CatMAP. None of the rest of the MBR results is significantly higher than the word-based result.

Overall, the Morfessor family of algorithms performed very well in both translation tasks. Categories-MAP was not as good as Morfessor Baseline or Allomorfessor, which is probably explained by the fact that it segmented words to shorter tokens. Also MetaMorph improved significantly the Finnish translations, but was not as useful in German.

5.4 Discussion

This was the first time that a SMT system was used to evaluate the quality of the morphological analysis. As the SMT tools applied are designed mostly for word-based translations, it was not a surprise that some problems arose.

Table 8. The BLEU results of the submitted unsupervised morpheme analyses used in SMT from **Finnish and German** for both Individual systems and MBR combination with word-based models (Competition 3)

Finnish-English				German-English			
Method	Comb.	Method	Indiv.	Method	Comb.	Method	Indiv.
MorfBase	0.2861	MorfBase	0.2742	MorfBase	0.3119	Allomorf	0.3001
Allomorf	0.2856	Allomorf	0.2717	Allomorf	0.3114	MorfBase	0.3000
gram-1	0.2821	MetaMorf	0.2631	gram-1	0.3103	CatMAP	0.2901
MetaMorf	0.2820	CatMAP	0.2610	P-Mimic	0.3086	gram-1	0.2873
CatMAP	0.2814	gram-1	0.2580	PM-Union	0.3083	MetaMorf	0.2855
PM-Union	0.2784	PM-Mimic	0.2347	PM-Mimic	0.3081	P-Mimic	0.2854
MorphNet	0.2779	P-Mimic	0.2252	CatMAP	0.3080	PM-Mimic	0.2821
PM-Mimic	0.2773	MorphNet	0.2245	MetaMorf	0.3077	MorphNet	0.2734
P-Mimic	0.2768	PM-Union	0.2223	MorphNet	0.3072	PM-Union	0.2729
		words	0.2764			words	0.3063

The word alignment tool used by the Moses system, Giza++, has strict limits on sentence lengths. A sentence cannot be longer than 100 tokens, and neither over 9 times longer or shorter than its sentence pair. Too long sentences are pruned away from the training data. Thus, the algorithms that segmented more, generally got less training data for the translation model. However, the dependency between average tokens per word and the amount of filtered training data was not linear. For example, the Morfessor CatMAP system could use much more training data than some of the algorithms that, on average, segmented less. Even without considering the decrease to the amount of training data available, oversegmentation is likely to be detrimental in the task, because it makes, e.g., the word alignment problem more complex. However, this sentence length restriction should be solved for the future evaluations.

After MBR combination, the rank of the algorithms was not the same as with the individual systems. Especially ParaMor-Morfessor Union system helped the word-based model more than its own BLEU score indicated. However, as the improvements were not statistically significant, the improved rank in the MBR combination may be affected more by just chance.

6 Conclusion

The Morpho Challenge 2009 was a successful follow-up to our previous Morpho Challenges 2005-2008. Since some of the tasks were unchanged from 2008, the participants of the previous challenges were able to track improvements of their algorithms. It also gave a possibility for the new participants and those who missed the previous deadlines to try more established benchmark tasks. New tasks were introduced for SMT which offer yet another viewpoint on what is required from morpheme analysis in practical applications.

The various evaluation results indicate the benefit of utilizing real-world applications for studying the morpheme analysis methods. Some algorithms that succeeded relatively well in imitating the grammatical morphemes did not perform as well in applications as others that differed more from the grammatical ones. Although the mutual performance differences of various algorithms in applications are often small, it seems that different applications may favor different kinds of morpheme, and thus, proposing the overall best morphemes is difficult.

Acknowledgments

We thank all the participants for their submissions and enthusiasm and the organizers of the PASCAL Challenge Program and CLEF who helped us to organize this challenge and its workshop. We are grateful to the University of Leipzig, University of Leeds, Computational Linguistics Group at University of Haifa, Stefan Bordag, Ebru Arisoy, Majdi Sawalha, Eric Atwell, and Mathias Creutz for making the data and gold standards in various languages available to the Challenge. This work was supported by the Academy of Finland in the project *Adaptive Informatics*, the graduate schools in Language Technology and Computational Methods of Information Technology, in part by the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022, and in part by the IST Programme of the European Community, under the FP7 project EMIME (213845) and PASCAL Network of Excellence. This publication only reflects the authors' views. We acknowledge that access rights to data and other materials are restricted due to other commitments.

References

1. Bilmes, J.A., Kirchhoff, K.: Factored language models and generalized parallel backoff. In: Proceedings of the Human Language Technology, Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), Edmonton, Canada, pp. 4–6 (2003)
2. Kurimo, M., Creutz, M., Varjokallio, M., Arisoy, E., Saraclar, M.: Unsupervised segmentation of words into morphemes - Challenge 2005, an introduction and evaluation report. In: PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes, Venice, Italy (2006)
3. Zieman, Y., Bleich, H.: Conceptual mapping of user's queries to medical subject headings. In: Proceedings of the 1997 American Medical Informatics Association (AMIA) Annual Fall Symposium (October 1997)
4. Kurimo, M., Creutz, M., Turunen, V.: Unsupervised morpheme analysis evaluation by IR experiments – Morpho Challenge 2007. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152. Springer, Heidelberg (2008)
5. Lee, Y.S.: Morphological analysis for statistical machine translation. In: Proceedings of the Human Language Technology, Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), Boston, MA, USA (2004)

6. Virpioja, S., Väyrynen, J.J., Creutz, M., Sadeniemi, M.: Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In: Proceedings of the Machine Translation Summit XI, Copenhagen, Denmark, pp. 491–498 (September 2007)
7. de Gispert, A., Virpioja, S., Kurimo, M., Byrne, W.: Minimum bayes risk combination of translation hypotheses from alternative morphological decompositions. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, Boulder, USA, Association for Computational Linguistics, pp. 73–76 (June 2009)
8. Kurimo, M., Virpioja, S., Turunen, V.T., Blackwood, G.W., Byrne, W.: Overview and results of Morpho Challenge 2009. In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece (2009)
9. Creutz, M., Lagus, K.: Inducing the morphological lexicon of a natural language from unannotated text. In: Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR 2005), Espoo, Finland, 106–113 (2005)
10. Creutz, M., Lagus, K.: Unsupervised discovery of morphemes. In: Proceedings of the Workshop on Morphological and Phonological Learning of ACL 2002, pp. 21–30 (2002)
11. Creutz, M., Lagus, K.: Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor. Technical Report A81, Publications in Computer and Information Science, Helsinki University of Technology (2005), http://www.cis.hut.fi/projects/morpho/
12. Sawalha, M., Atwell, E.: Comparative evaluation of arabic language morphological analysers and stemmers. In: Proceedings of COLING 2008 22nd International Conference on Computational Linguistics (2008)
13. Kurimo, M., Creutz, M., Varjokallio, M.: Unsupervised morpheme analysis evaluation by a comparison to a linguistic Gold Standard – Morpho Challenge 2007. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152. Springer, Heidelberg (2008)
14. Kurimo, M., Varjokallio, M.: Unsupervised morpheme analysis evaluation by a comparison to a linguistic Gold Standard – Morpho Challenge 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706. Springer, Heidelberg (2009)
15. Kurimo, M., Turunen, V.: Unsupervised morpheme analysis evaluation by IR experiments – Morpho Challenge 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706. Springer, Heidelberg (2009)
16. Creutz, M., Linden, K.: Morpheme segmentation gold standards for finnish and english. Technical Report A77, Publications in Computer and Information Science, Helsinki University of Technology (2004), http://www.cis.hut.fi/projects/morpho/
17. Hull, D.A.: Using statistical testing in the evaluation of retrieval experiments. In: SIGIR 1993: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 329–338. ACM Press, New York (1993)
18. Agirre, E., Di Nunzio, G.M., Ferro, N., Mandl, T., Peters, C.: CLEF 2008: Ad hoc track overview. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 15–37. Springer, Heidelberg (2009)

19. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: Proceedings of the 10th Machine Translation Summit, Phuket, Thailand, pp. 79–86 (2005)
20. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Annual Meeting of ACL, Demonstration Session, Czech Republic (June 2007)
21. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002), Morristown, NJ, USA, pp. 311–318. Association for Computational Linguistics (2002)
22. Kumar, S., Byrne, W.: Minimum Bayes-Risk decoding for statistical machine translation. In: Proceedings of Human Language Technologies: The 2004 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 169–176 (2004)
23. Tromble, R., Kumar, S., Och, F., Macherey, W.: Lattice Minimum Bayes-Risk decoding for statistical machine translation. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, pp. 620–629. Association for Computational Linguistics (October 2008)
24. Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., Mohri, M.: OpenFst: A general and efficient weighted finite-state transducer library. In: Holub, J., Žďárek, J. (eds.) CIAA 2007. LNCS, vol. 4783, pp. 11–23. Springer, Heidelberg (2007)
25. Sim, K.C., Byrne, W.J., Gales, M.J.F., Sahbi, H., Woodland, P.C.: Consensus network decoding for statistical machine translation. In: IEEE Conference on Acoustics, Speech and Signal Processing (2007)

MorphoNet: Exploring the Use of Community Structure for Unsupervised Morpheme Analysis

Delphine Bernhard*

ILES group, LIMSI-CNRS, Orsay, France
delphine.bernhard@limsi.fr

Abstract. This paper investigates a novel approach to unsupervised morphology induction relying on community detection in networks. In a first step, morphological transformation rules are automatically acquired based on graphical similarities between words. These rules encode substring substitutions for transforming one word form into another. The transformation rules are then applied to the construction of a lexical network. The nodes of the network stand for words while edges represent transformation rules. In the next step, a clustering algorithm is applied to the network to detect families of morphologically related words. Finally, morpheme analyses are produced based on the transformation rules and the word families obtained after clustering. While still in its preliminary development stages, this method obtained encouraging results at Morpho Challenge 2009, which demonstrate the viability of the approach.

1 Introduction

Unsupervised morphology induction, which is the goal of the Morpho Challenge competition series [1], consists in automatically discovering a word's morphemes using only minimal resources such as a list of the words in the target language and a text corpus. Ideally, unsupervised algorithms should be able to learn the morphology of a large variety of languages; for Morpho Challenge 2009, the target languages were English, Finnish, German, Turkish and Arabic.

For our participation at Morpho Challenge 2009 we developed a novel method for unsupervised morphology induction called *MorphoNet*. MorphoNet relies on a *network* representation of morphological relations between words, where nodes correspond to whole word forms and edges encode morphological relatedness. Networks have been successfully used in recent years to represent linguistic phenomena for tasks such as word clustering [2], word sense disambiguation [3], summarisation, or keyword extraction [4]. Moreover, network-based methods have been shown to perform well for a wide range of NLP applications. In line with this body of research, we propose to represent morphological phenonema as a network. This approach has two major advantages. First, it is theoretically related to linguistic theories such as the Network Model by J. Bybee [5] or

* This work was done while the author was at the UKP Lab, Darmstadt, Germany.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 598–608, 2010.

whole word morphology [6]. It differs from traditional linear concatenative approaches to morphology in that words are not explicitely split into a sequence of morphemes, but related to one another through morphological transformations. Second, it enables the use of effective network-based clustering and ranking methods. Our model thus benefits from research done on graphs in other domains such as sociology [7] or other areas of NLP. We especially investigate the use of *community structure* for morphology induction. Networks with community structure contain groups of nodes with dense interconnections; in our case, communities correspond to families of morphologically related words. Communities can be automatically identified in networks with community detection algorithms. To our knowledge, this is the first time that community detection algorithms are applied to the task of unsupervised morphology induction.

Though in its very early development stages, the approach yields promising results at Morpho Challenge 2009 when compared to standard baselines such as the Morfessor algorithms [8, 9].

The article is structured as follows. In the next section, we report related work. Then, we describe our method for building lexical networks. In Sect. 4, we explain how word families can be discovered based on the network structure, while in Sect. 5 we detail our approach for obtaining morpheme analyses. Evaluation results are given in Sect. 6.

2 Related Work on Morphology Induction

Morphological analysis is useful for many applications like speech recognition and synthesis, machine translation or information retrieval. However, all these applications of morphology necessitate morphological resources which are not available for all languages, or, when available, are often incomplete. Much research has therefore been devoted to the unsupervised acquisition of morphological knowledge.

Methods for the unsupervised acquisition of morphological knowledge can be classified according to the intended result: (i) identification of morphologically related words (*clustering*), (ii) splitting of words into morphs (*segmentation*), and (iii) identification of morphemes (*analysis*). Morpheme analysis is the goal of the latest Morpho Challenge competitions, while for some applications, such as information retrieval, it is often sufficient to retrieve morphologically related words without proceeding to a full analysis. The identification of morphologically related words has been attempted by unsupervised methods [10] as well as approaches using dictionaries as input data [11].

Segmentation is certainly the method which has gathered the largest amount of interest in the NLP research community [8, 12–14]. It follows linear concatenative approaches to morphology such as item-and-arrangement, which postulates that words are formed by putting morphemes together. There are, however, some well known limitations to purely concatenative approaches, which are seldom dealt with by unsupervised segmentation methods. These limitations include ablaut, umlaut, and infixation. Contrarily to unsupervised morpheme segmentation methods, MorphoNet makes no assumption on the internal structure and

morphotactics of words. It identifies flexible word transformation rules which encode substring substitutions for transforming one word form into another. These transformation rules are not limited to concatenative processes such as prefixation or suffixation (see Sect. 3.2) and thus aim at addressing some of the limitations of concatenative approaches.[1]

Unsupervised methods rely on many properties for morphology induction, which are too numerous to be listed here. The most obvious cue is usually *graphical relatedness*: two words which share a long enough common substring are likely to be morphologically related. Graphical relatedness can be estimated by measures of orthographic distance [15] or by finding the longest initial (or final) substring [16, 17]. Our system is related to these methods in that it uses fuzzy string similarity to bootstrap the morphology induction process.

3 Lexical Networks

3.1 Use of Graphs for Morphology Induction

A network can be mathematically represented as a graph. Formally, a graph G is a pair (V, E), where V is a set of vertices (nodes) and $E \subseteq V \times V$ is a set of edges (lines, links). The main advantage of graphs is that they make it possible to take into account multiple dependencies across elements, so that the whole network plays an important role on the results obtained for a single element.

The lexical networks built by our method consist of word nodes linked by edges which encode morphological relations. Similar lexical networks have been previously described by Hathout [18]. Our approach differs however from Hathout's in two main aspects: (i) it uses only a raw list of words as input, while Hathout's method acquires morphological links from WordNet, and (ii) we attempt to take a broader range of morphological phenomena into account by acquiring morphological transformation rules which are not limited to suffixation.

3.2 Acquisition of Morphological Transformation Rules

The first step in our method consists in acquiring a set of *morphological transformation rules*. Morphological transformation rules make it possible to transform one word into another by performing substring substitutions. We represent a rule R with the following notation: `pattern` → `repl`, where `pattern` is a regular expression and `repl` is the replacement with backreferences to capturing groups in the pattern. For instance, the rule `^(.+)ly$` → `\1` applies to the word *totally* to produce the word *total*.

Tranformation rules are in principle not limited to concatenative processes, which should be especially useful for languages such as Arabic, e.g. when inducing rules for word pairs such as *kataba* (he wrote) and *kutiba* (it was written).

[1] However, MorphoNet does not address cases of morphologically related words with no orthographic overlap, such as *be* and *was*.

These rules are acquired using a subset L of the wordlist W provided for each language to avoid noise given the substantial length of the word lists provided for MorphoChallenge. In our experiments, we used the 10,000 most frequent words whose length exceeds the average word (type) length.[2] The method used to acquire the rules is described in detail in Algorithm 1.

Algorithm 1. Procedure for the acquisition of morphological transformation rules, given an input list of words L

```
1: rules ← ∅
2: n ← len(L)
3: for i = 1 to n do
4:    w ← L[i]
5:    matches ← get_close_matches(w, L[i + 1 : n])
6:    for w2 in matches do
7:       r ← get_rule_from_word_pair(w, w2)
8:       add r to rules
9:    end for
10: end for
11: return rules
```

For each word w in the list L we retrieve graphically similar words (Line 5, get_close_matches) using a *gestalt* approach to fuzzy pattern matching based on the Ratcliff-Obershelp algorithm.[3] This string comparison method computes a measure of the similarity of two strings relying on the number of characters in matching subsequences. For example, given the target word *democratic*, the following close matches are obtained: *undemocratic*, *democratically*, *democrats*, *democrat's*, *anti-democratic*. We then obtain rules (Line 7, get_rule_from_word_pair) by comparing the target word with all its close matches and identifying the matching subsequences;[4] for instance given the word *democratic* and its close match *undemocratic*, we obtain the following rule: `^un(.+)$` → `\1`.

We have kept all rules which occur at least twice in the training data.[5] Moreover, no attempt is made to distinguish between inflection and derivation.

Table 1 lists the number of transformation rules obtained from the datasets provided for Morpho Challenge 2009[6] along with some examples:

3.3 Construction of a Lexical Network

Once transformation rules have been acquired, they are used to build a lexical network represented as a graph. Nodes in the graph represent words from the

[2] Except for Arabic, where there are only 9,641 word forms which are longer than the average word length in the vowelized version and 6,707 in the non-vowelized version.

[3] We used the implementation provided by the Python *difflib* module with the cutoff argument set to 0.8.

[4] Matching subsequences are identified by the get_matching_blocks Python method.

[5] For Arabic, we even kept all rules given the small size of the input word list.

[6] http://www.cis.hut.fi/morphochallenge2009/

Table 1. Morphological transformation rules acquired for the input datasets

Language	# rules	Examples	
English	834	`^re(.+)s$ → \1`	`^(.+)'s$ → \1`
Finnish	1,472	`^(.+)et$ → \1ia`	`^(.+)ksi$ → \1t`
German	771	`^(.+)ungen$ → \1t`	`^(.+)ge(.+)t$ → \1\2en`
Turkish	3,494	`^(.+)n(.+)$ → \1\2`	`^(.+)nde$ → \1`
Arabic vowelized	8,974	`^(.+)iy(.+)$ → \1uw\2`	`^(.+)K$ → \1N`
Arabic non-vowelized	2,174	`^(.+)wA$ → \1`	`^(.+)hm$ → \1`

input word list W. Two words w_1 and w_2 are connected by an edge if there exists a transformation rule R such that $\mathrm{R}(w_1) = w_2$. The graph obtained using this method is directed based on the direction of the rules applied. Figure 1 displays an example lexical network.

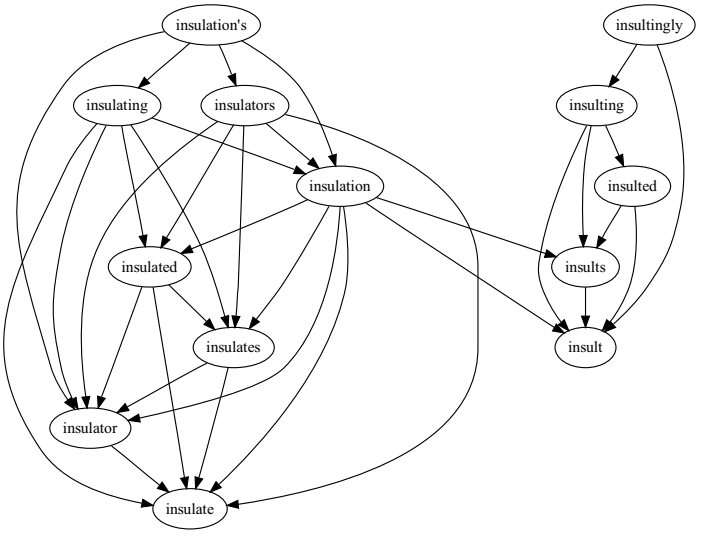


Fig. 1. Example lexical network

4 Acquisition of Word Families

The graphs we obtain usually contain one large connected component, along with smaller connected components. Extracting connected components is thus not reliable enough to identify word families, i.e. groups of words which are related both semantically and orthographically. For instance, the lexical network depicted in Fig. 1 contains one large connected component, which clearly consists of two different word families. The induction of word families can be formulated as a classical problem of community detection in graphs. Communities

correspond to groups of tightly-knit nodes characterised by a high intra-group density and a lower inter-group density [19]. There are several methods to detect communities in graphs. Markov Clustering [20] for instance consists in partitioning a graph by simulating random walks in the graph; it has been used to detect communities in a graph of nouns by Dorow et al. [21]. The community detection method described by Newman [19] has been applied to natural language data by Matsuo et al. [2] for graphs based on word similarity measures by web counts.

The method proposed by Newman relies on the modularity function Q which measures the quality of a division of a graph into communities. The advantages of this method are that it is not necessary to know the number of communities beforehand and it needs no fine parameter tuning. Modularity compares the number of edges within communities to the number of expected edges:

$$Q = \sum_i (e_{ii} - (\sum_j e_{ij})^2) \tag{1}$$

where e_{ii} is the fraction of the edges in the network that connect nodes within community i, e_{ij} is one-half of the fraction of edges in the network that connect nodes in community i to those in community j and $\sum_j e_{ij}$ is the fraction of edges connected to nodes in community i.

A good division corresponds to more edges within communities than would be expected by random chance, that is a positive modularity value Q. Modularity is high when there are many edges within communities and few between them. By applying Newman's algorithm on the lexical network of Fig. 1, two communities are identified: {*insulation's, insulating, insulators, insulation, insulated, insulates, insulator, insulate*} and {*insultingly, insulting, insulted, insults, insult*}.

The main difficulty lies in finding the division which yields the best value for Q. It is of course infeasible to test each possible division of the network. Newman [19] therefore proposes a method of agglomerative hierarchical clustering starting from communities made of a single node. Communities are repeatedly joined together in pairs, choosing the join that leads to the biggest increase (or slightest decrease) of Q. The best partition of the network in communities corresponds to the biggest value of Q.

Our experiments with the Newman Clustering algorithm have nevertheless shown that it tends to detect bigger communities than wanted, thus decreasing the precision. We have therefore added an additional constraint on possible joins by measuring the density of edges across communities (*cross-community edge density*).

Cross-community edge density between communities A and B is defined as follows:

$$D_{AB} = \frac{\mathsf{number\_of\_edges}(A, B)}{|A| \times |B|} \tag{2}$$

where $\mathsf{number\_of\_edges}(A, B)$ is the number of edges linking nodes in community A to nodes in community B, and $|A|$ and $|B|$ are the number of nodes in community A and B, respectively. The minimum cross-community edge density is fixed by a parameter d whose value ranges from 0 to 1.

5 Morpheme Analyses

After performing clustering, morpheme analyses are obtained based on the word families identified and the transformation rule edges linking words which belong to the same family. First, a representative word is identified for each word family: this is the shortest word in the family; in case of a tie, the most frequent among the shortest words is chosen. The full morpheme analysis for a word form w consists of its family representative and a string representation of the transformation rules that apply to w. The method is detailed in Algorithm 2.

Algorithm 2. Procedure for obtaining the morpheme analyses, given a word family C and the lexical network G

```
analyses[*] ← ∅
subg ← get_subgraph(G, C)
for edge (w1, w2, rule) in subg do
    analyses[w1] ← analyses[w1] ∪ to_plain_string(rule.pattern)
    analyses[w2] ← analyses[w2] ∪ to_plain_string(rule.repl)
end for
rep ← get_family_representative(C)
for word w in word family C do
    analyses[w] ← analyses[w] ∪ rep
end for
return analyses
```

Example 1. Consider for instance the communities obtained for Fig. 1. The representative for the word family {*insulted*;*insulting*;*insult*;*insults*;*insultingly*} is *insult* since it is the shortest word. The complete analyses for the words are the following:

`insultingly`	`insult _ly _ingly`
`insulting`	`insult _ing`
`insulted`	`insult _ed`
`insults`	`insult _s`
`insult`	`insult`

Two transformation rules apply to the word *insultingly*: `^(.+)ly$` → `\1` and `^(.+)ingly$` → `\1`, which are represented in the final analysis as `_ly _ingly`.

6 Evaluation

In this section, we report the results obtained by MorphoNet at Morpho Challenge 2009 competition 1 (linguistic evaluation) and 2 (information retrieval). For all languages, the value of parameter d (cross-community edge density) was empirically set to 0.1 for community detection.

6.1 Morpho Challenge Competition 1

Table 2 contains the results of the linguistic evaluation (competition 1) for MorphoNet and a simple reference method consisting in splitting words into letters. Results are measured in terms of Precision, Recall and F-Measure.

Table 2. Results for competition 1

Language	Method	Precision	Recall	F-Measure	Rank
English	MorphoNet	65.08%	47.82%	55.13%	7 / 14
	letters	3.82%	99.88%	7.35%	
German	MorphoNet	67.41%	30.19%	41.71%	8 / 15
	letters	2.79%	99.92%	5.43%	
Finnish	MorphoNet	63.35%	22.62%	33.34%	9 / 12
	letters	5.17%	99.89%	9.83%	
Turkish	MorphoNet	61.75%	30.90%	41.19%	7 / 14
	letters	8.66%	99.13%	15.93%	
Arabic vowelized	MorphoNet	92.52%	2.91%	5.65%	8 / 12
	letters	50.56%	84.08%	63.15%	
Arabic non vowelized	MorphoNet	90.49%	4.95%	9.39%	6 / 12
	letters	70.48%	53.51%	60.83%	

MorphoNet performs best for English. The lowest results are obtained for Arabic, which is characterized by good precision but very low recall. Most of the participating systems obtained comparably low results for Arabic and none was able to beat the "letters" reference. This could be explained by the following reasons: (i) the datasets provided for Arabic were too small for MorphoNet to perform well and (ii) the analyses required for Arabic were far more complex (in terms of the number of morphemes per word) than for the other languages.

The results also show that MorphoNet consistently obtains better precision than recall, especially in Arabic. The method relies on a list of transformation rules which are automatically acquired in a first step. It is therefore likely that some important rules are missing, leading to low recall. This problem might be solved by performing multiple iterations of rule induction and clustering or by applying rules in a cascaded manner, so that one rule applies to the output of another rule.

Moreover, the procedure for obtaining morpheme analyses is still very coarse, leading to composite morphemes such as `_ingly`. This could easily be improved by detecting and further decomposing such morphemes.

Finally, transformation rules could be weighted by their productivity or their frequency to improve clustering, since some transformation rules might be more reliable than others.

6.2 Morpho Challenge Competition 2

Table 3 summarises the results of the information retrieval (IR) task (competition 2). Results without morpheme analysis (no analysis) are also provided.

Table 3. IR results (mean average precision MAP)

Method	English	German	Finnish
MorphoNet	0.3560	0.3167	0.3668
No analysis	0.3293	0.3509	0.3519

MorphoNet improves the IR results over unanalysed words for English and Finnish, but not for German. While it is difficult to come up with a clear explanation, this might be due to the compounding nature of German. Indeed, the MorphoNet system does not directly cope with compounding for the time being, which might be especially detrimental to the IR task.

7 Conclusions and Future Work

We have described a novel linguistically motivated approach to unsupervised morpheme analysis relying on a network representation of morphological relations between words. Due to the underlying network representation, it is possible to use community detection and ranking methods devised for other kinds of data. This approach is still in its very early stage, yet the results obtained at Morpho Challenge 2009 demonstrate that it yields very promising results and thus deserves further investigation.

The method described in this paper can be considered as a baseline for network-based morphology induction. It leaves lots of room for improvement. A first objective would be to obtain a better recall for morpheme analysis. This necessitates to provide a better mechanism for the acquisition of transformation rules. It should be possible to perform multiple iterations of the rule induction and clustering cycle or to apply rules in a cascaded manner. This is especially needed for languages which are morphologically more complex than English such as Turkish or Finnish. Also, we have not weighted the edges in the graph, which could be useful to improve clustering.

The clustering method performs hard-clustering: each word belongs to only one family. This is especially detrimental for languages like German, for which it would be desirable to allow multiple family membership in order to take compounding into account. In the future, we would therefore like to better address compounding.

Graphs also open up the way for a new form of modelisation of morphology enabling the analysis of crucial morphological properties. In particular, node properties in the graph could be used to rank nodes and better detect base words in families, using algorithms such as PageRank.

Acknowledgements

We are grateful to the anonymous reviewers for their helpful comments.

References

1. Kurimo, M., Virpioja, S., Turunen, V.T., Blackwood, G.W., Byrne, W.: Overview and Results of Morpho Challenge 2009. In: Multilingual Information Access Evaluation 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Revised Selected Papers. LNCS, vol. I. Springer, Heidelberg (2010)
2. Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M.: Graph-based Word Clustering using a Web Search Engine. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 542–550 (2006)
3. Mihalcea, R.: Unsupervised Large-Vocabulary Word Sense Disambiguation with Graph-based Algo-rithms for Sequence Data Labeling. In: Proceedings of the HLT/EMNLP 2005 Conference, pp. 411–418 (2005)
4. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of EMNLP 2004, pp. 404–411 (2004)
5. Bybee, J.: Morphology: A Study of the Relation between Meaning and Form. Benjamins, Philadelphia (1985)
6. Neuvel, S., Fulop, S.A.: Unsupervised Learning of Morphology Without Morphemes. In: Proceedings of the ACL Workshop on Morphological and Phonological Learning 2002, pp. 31–40 (2002)
7. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69 (2004)
8. Creutz, M., Lagus, K.: Unsupervised Discovery of Morphemes. In: Proceedings of the ACL Workshop on Morphological and Phonological Learning 2002, pp. 21–30 (2002)
9. Creutz, M., Lagus, K.: Inducing the Morphological Lexicon of a Natural Language from Unannotated Text. In: Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning, AKRR 2005 (2005)
10. Bernhard, D.: Apprentissage non supervisé de familles morphologiques par classification ascendante hiérarchique. In: Actes de la 14e conférence sur le Traitement Automatique des Langues Naturelles TALN 2007, pp. 367–376 (2007)
11. Hathout, N.: Acquistion of the Morphological Structure of the Lexicon Based on Lexical Similarity and Formal Analogy. In: Proceedings of the 3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing (COLING 2008), pp. 1–8 (2008)
12. Bernhard, D.: Unsupervised Morphological Segmentation Based on Segment Predictability and Word Segments Alignment. In: Proceedings of the Pascal Challenges Workshop on the Unsupervised Seg-mentation of Words into Morphemes, pp. 19–23 (April 2006)
13. Dasgupta, S., Ng, V.: High-Performance, Language-Independent Morphological Segmentation. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007), pp. 155–163 (2007)
14. Demberg, V.: A Language-Independent Unsupervised Model for Morphological Segmentation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 920–927 (2007)
15. Baroni, M., Matiasek, J., Trost, H.: Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In: Proceedings of the ACL Workshop on Morphological and Phonological Learning 2002, pp. 48–57 (2002)

16. Gaussier, E.: Unsupervised learning of derivational morphology from inectional lexicons. In: Proceedings of the Workshop on Unsupervised Methods in Natural Language Processing (1999)
17. Schone, P., Jurafsky, D.: Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. In: Proceedings of the Fourth Conference on Computational Natural Language Learning, pp. 67–72 (2000)
18. Hathout, N.: From WordNet to CELEX: acquiring morphological links from dictionaries of synonyms. In: Proceedings of the Third International Conference on Language Resources and Evaluation, pp. 1478–1484 (2002)
19. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E 69 (2004)
20. van Dongen, S.: Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht (2000)
21. Dorow, B., Widdows, D., Ling, K., Eckmann, J.P., Sergi, D., Moses, E.: Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination. In: 2nd MEANING Workshop (2005)

Unsupervised Morpheme Analysis with Allomorfessor

Sami Virpioja, Oskar Kohonen, and Krista Lagus

Adaptive Informatics Research Centre
Aalto University School of Science and Technology
{sami.virpioja,oskar.kohonen,krista.lagus}@tkk.fi

Abstract. Allomorfessor extends the unsupervised morpheme segmentation method Morfessor to account for the linguistic phenomenon of allomorphy, where one morpheme has several different surface forms. The method discovers common base forms for allomorphs from an unannotated corpus by finding small modifications, called mutations, for them. Using Maximum a Posteriori estimation, the model is able to decide the amount and types of the mutations needed for the particular language. In Morpho Challenge 2009 evaluations, the effect of the mutations was discovered to be rather small. However, Allomorfessor performed generally well, achieving the best results for English in the linguistic evaluation, and being in the top three in the application evaluations for all languages.

1 Introduction

Many successful methods for unsupervised learning of morphology are based on the *segmentation* of word forms into smaller, morpheme-like units (see, e.g., [1,4,6]). Although the segmentation approach is reasonable for many languages with mostly concatenative morphology, it does not model the phenomenon of *allomorphy*. Allomorphs are different morph-level surface realizations of an underlying morpheme-level unit, of which only one may appear in a certain morpho- and phonotactical context. For example, in English, `pretty` has an allomorph `pretti` as in `pretti+er`. Similarly for German `Haus` (house) – `Häus+er` (houses), and for Finnish `kenkä` (shoe) – `kengä+n` (shoe's).

We focus on the problem of unsupervised learning of a morpheme analyzer, in contrast to finding morphologically related words or morphological segmentations. A limitation in most of the suggested methods for this task is that they do not model general concatenative morphology, including, e.g., compound words. Either only stem-suffix pairs are considered [11], or there can be multiple suffixes but only one stem [2]. In [5], many stems are allowed, but the approach cannot find, e.g., affixes between stems. In contrast, our work [7,10] combines the learning of allomorphy to the modeling of general concatenative morphology. The model, called Allomorfessor, is based on Morfessor Baseline [3,4]. In Allomorfessor, a word is assumed to consist of a sequence of one or more morphemes and mutations that transform the common base form into the different allomorphs. In this article, we describe our methodology for modeling allomorphic variations

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 609–616, 2010.

in stems, the current probabilistic model, and the algorithms for learning the model parameters and applying the model for analyzing new words. In addition, we discuss our results in the Morpho Challenge 2009 tasks.

2 Modeling Allomorphy with Mutations

We model allomorphy with orthographic transformations that make minor modifications to the surface forms of the morphemes. We call these transformations *mutations*. The choice for the class of mutations is not trivial: The mutations should be general enough so that similar variations in different stems can be modeled with the same mutation, but they should also be restricted so that the number of wrong analyses becomes as small as possible. We apply mutations that consist of a sequence of substitution and deletion operations (Table 1), which are applied in order, starting from the end of the stem. Each operation modifies the k:th letter that is equal to the target letter of the operation, starting from the current position. The smallest mutation between two arbitrary strings is trivial to find with a dynamic programming algorithm.

Table 1. The allowed operations in mutations and some examples in Finnish

Operation	*Notation*	*Description*
substitution	*k*`x|y`	Change *k*:th `x` to `y`
deletion	`-`*k*`x`	Remove *k*:th `x`

(k is omitted when $k = 1$)

Source	*Mutation*	*Target*
`kenkä` (shoe)	`(k|g)`	`kengä` (e.g. `kengä+ssä`, in a shoe)
`tanko` (pole)	`(k|g)`	`tango` (e.g. `tango+t`, poles)
`ranta` (shore)	`(-a t|n)`	`rann` (e.g. `rann+oi+lla`, on shores)
`ihminen` (human)	`(2n|s)`	`ihmisen` (human's)

To make the approach more feasible, we attempt to consider mutations only in stems. In practice, we do not allow mutations to the last morph of a word form, nor to any morph having a length less than four characters. For short morphs, such as typical affixes, even a minor modification is likely to change a morpheme into a different one (consider, e.g., prefixes `re` and `pre`).

To get an estimate of how much these restrictions limit the ability to model allomorphic variation, we calculated to which extent the allomorphs in a linguistic gold standard segmentation could be described using such mutations. We calculated statistics separately for morph types and morph tokens in the corpus using the Morpho Challenge 2009 data sets for English, Finnish, and Turkish. As shown in Table 2, mutations provide a reasonably good coverage for English and Finnish. E.g., for English, we can describe 82% of the real allomorphs in the gold standard segmentation. The percentages for the corpora are lower, as affixes are more frequent than stems. For Turkish, where most of the allomorphy seems

to be in affixes or other short morphemes, only 2% of the cases with allomorphic variants in a corpus can be found using the type of mutations that we apply.

Table 2. The proportion of morphs with allomorphic variation and how many variants can be modeled with mutations for English, Finnish, and Turkish. The first column shows the number of morphs in the data, the second column shows how many of the morphs are surface realizations of a morpheme with allomorphic variation, and the third column shows how many of the allomorphs can be constructed with mutations.

	Morphs	*Allomorphs*	*Mutation found*
English types	21 173	10 858 (51.3%)	8 912 (82.1%)
Finnish types	68 743	56 653 (82.4%)	36 210 (63.9%)
Turkish types	23 376	646 (2.8%)	102 (15.8%)
English tokens	76 968 382	42 282 837 (54.9%)	14 706 543 (34.8%)
Finnish tokens	73 512 023	61 583 251 (83.8%)	18 751 022 (30.5%)
Turkish tokens	23 288 821	11 978 142 (51.4%)	225 708 (1.9%)

3 Allomorfessor Model and Algorithms

We learn morphology using a probabilistic model that assumes that the observed word lexicon $\mathcal{L}_W$ is generated from a set of morphs that can be modified by mutations. Our model is an extension to Morfessor Baseline [3], for which the *Maximum a Posteriori* (MAP) formulation can be expressed as:

$$\mathcal{M}_{\text{MAP}} = \arg\max_{\mathcal{M}} P(\mathcal{M}|\mathcal{L}_W) = \arg\max_{\mathcal{M}} P(\mathcal{M})P(\mathcal{L}_W|\mathcal{M}) \quad (1)$$

$$P(\mathcal{L}_W|\mathcal{M}) = \prod_{j=1}^{W} \prod_{\mu_{jk} \in z_j} P(\mu_{jk}), \quad (2)$$

where μ_{jk} is the k:th morph in w_j. We assume that the observed W word forms are independent. Furthermore, the probability of each morph, given by $P(\mu)$, is assumed to be independent. The hidden variables z_j give the analysis of the word forms w_j. Each z_j is a list of substrings specifying one of the non-overlapping segmentations of the word form into morphs. $P(\mathcal{M})$ is the prior probability for the model. It is derived using the Minimum Description Length (MDL) principle [9] and it controls the model complexity, for example, the amount of morphs.

In Allomorfessor, we extend the analyses z_j so that each substring is produced by applying the mutation $\delta_{j(k+1)}$ to the previous morph μ_{jk}. I.e., if s is the start and t is the end position of the surface form of the k:th morph of w_j, there is a constraint $\delta_{j(k+1)}(\mu_{jk}) = w_{js..t}$, where $w_{js..t}$ is the corresponding substring. The mutation may be empty, denoted ϵ, such that $\epsilon(\mu) = \mu$. The mutation in the first position of an analysis is always empty, as it cannot be applied to any morph. For example, if the word w_j is `prettier`, a possible analysis is $z_j = [(\epsilon, \texttt{pretty}), (\texttt{(y|i)}, \texttt{er})]$. Assuming that the morphs and mutations are independent leads to

severe undersegmentation [7], and therefore we condition the probability of the mutation δ_{jk} on the following morph μ_{jk}. This results in the likelihood:

$$P(\mathcal{L}_W|\mathcal{M}) = \prod_{j=1}^{W} \prod_{(\delta_{jk},\mu_{jk}) \in z_j} P(\delta_{jk}|\mu_{jk})P(\mu_{jk}) \quad (3)$$

The prior $P(\mathcal{M})$, described in detail in [10], is very similar to that of Morfessor Baseline. For mutations, the prior follows an MDL-based derivation similar to morphs. Informally, the fewer different morph or mutation types there are, the higher the prior probability is. Moreover, shorter morphs and mutations have higher probabilities than longer ones.

3.1 Algorithm for Learning Model Parameters

For learning the model parameters, we apply an iterative greedy algorithm similar to the one in Morfessor Baseline [3]. The algorithm finds an approximate solution for the MAP problem in Equation 1. Initially, the set of morphs contains all the words in the corpus and only an empty mutation exists. In one epoch, all words are picked in a random order. For each word w_j, different analyses z_j are considered, and the probabilities $P(\delta_{jk}|\mu_{jk})$ and $P(\mu_{jk})$ are updated to their Maximum Likelihood parameters corresponding to the analysis z_j. The analysis z_j^* that maximizes the posterior probability $P(\mathcal{M}|\mathcal{L}_W)$ is then selected. The algorithm considers analyzing the word w (1) without splits, (2) with all possible two-way splits of w and an empty mutation, and (3) with all possible two-way splits where the prefix part is modified by a non-empty mutation. If the selected analysis includes a split, the prefix and suffix are analyzed recursively. To make the approach computationally feasible, heuristic restrictions are applied when testing splits with non-empty mutations: The morph and its potential base form have to begin with the same letters, the base form has to occur as a word, and the suffix has to be already in the lexicon. Finally, only K analyses per morph are tested, which results in time complexity $O(KW \log(W))$ for one epoch of the learning algorithm. Pseudocode for the algorithm is presented in [7] and [10].

3.2 Algorithm for Analyzing New Data

After training a model, it can be used to analyze words with a variant of the *Viterbi algorithm*, a dynamic programming algorithm that finds the most probable state sequences for Hidden Markov Models (HMM). Here, the observation is the sequence of $|w|$ letters that form the word w and the hidden states are the morphemes of the word. Unlike standard HMMs, the states can emit observations of different lengths. We need a grid s of length $|w|$ to fill with the best unnormalized probability values $\alpha(s_i)$ and paths. Without mutations, the model is 0th order Markov model and the grid is a one dimensional table. The grid position s_i indicates that the first i letters are observed. At each time step, we proceed with one letter and insert the value $\alpha(s_i) = \max_j \alpha(s_j)P(\mu_{ji})$ and path

indicator $\psi(s_i) = \arg\max_j \alpha(s_j)P(\mu_{ji})$ to the grid. We can arrive at s_i from any of the positions s_j between s_1 and s_{i-1}: The letters between j and i form the next morpheme μ_{ji}. The time complexity of the algorithm is thus $O(|w|^2)$.

With mutations, the algorithm becomes a bit more complicated. As mutations are conditioned on the suffixes, it is easier to run the algorithm from right to left. The grid has to be two dimensional: For each s_i there can be several states (morphemes) with their own costs and paths. The rule for updating the grid value for s_i is

$$\alpha(s_i, \hat{\mu}_{ij}) = \max_{j \in [i+1, |w|]} \Big\{ \max_{\mu \in s_j} \Big\{ \max_{\delta \in \Delta} \big\{ \alpha(s_j, \mu) P(\delta|\mu) P(\hat{\mu}_{ij}) \big\} \Big\} \Big\}, \tag{4}$$

where $\hat{\mu}_{ij}$ is a morpheme that produces the letters between i and j when modified by the mutation δ. Only those mutations Δ that are observed before μ need to be tested, otherwise $P(\delta|\mu) = 0$. For the morphemes that are not observed before, we use an approximate cost of adding them into the lexicon. The worst case time complexity for the algorithm is $O(MD|w|^2)$. In practice, however, the number of morphemes M and mutations D tested in each position is quite limited.

4 Experiments and Discussion

The algorithm was evaluated in Morpho Challenge 2009. Details of the competitions and results are presented in [8]. For Competitions 1 and 2 we trained the model with the Competition 1 data, where all words occurring only once were filtered out to remove "noisy" data such as misspelled words, with the exception of the very small Arabic data sets. The training algorithm converged in 5–8 epochs and the total training time varied from ten minutes (Arabic) to one week (Finnish). After training the model, we analyzed all the words with the Viterbi algorithm. For Competition 3, we used the Europarl data set for training without any filtering and then applied the Viterbi algorithm. In all models, the following priors and parameter settings were used: Morpheme length distribution was geometric with the parameter $p = W/(W + C)$, where W is the number of words and C is the number of characters in the training corpus. For mutation lengths, we used a gamma distribution with parameters equal to one. The number of candidates K considered for each morph during the training was 20.

In Competition 1, the algorithms were compared to a linguistic gold standard analysis and scored according to F-measure, i.e., the geometric mean of precision and recall. For English, Allomorfessor achieved the winning F-measure. For the other languages, the results were only moderate due to the low recall: All methods that outperformed Allomorfessor had a higher recall. In Competition 2, the algorithms were applied in an information retrieval system for English, Finnish and German. There, Allomorfessor performed reasonably well, being second in English and Finnish and third in German. In Competition 3, where the algorithms were applied to Finnish-to-English and German-to-English machine translation systems, Allomorfessor was the best of the submitted methods.

As Allomorfessor is very close to Morfessor Baseline, we have to compare them to see the effect of mutations. In Table 3, the performance of the current

Table 3. Evaluation results for different versions of Allomorfessor and Morfessor. For Competition 1 results (C1), precision, recall and F-measure are given. C2 is scored with mean average precision (MAP) and C3 with BLEU. The methods marked with an asterisk were trained with data where words occurring only once were excluded.

	Allomorfessor Alpha (-08)	Allomorfessor Baseline*	Morfessor Baseline*	Morfessor Baseline	Morfessor CatMAP
English					
C1 precision	83.31%	68.98%	68.43%	74.93%	84.75%
C1 recall	15.84%	56.82%	56.19%	49.81%	35.97%
C1 F-measure	26.61%	**62.31%**	61.71%	59.84%	50.50%
C2 MAP	-	38.52%	38.73%	**46.56%**	37.13%
Finnish					
C1 precision	92.64%	86.51%	86.07%	89.41%	79.01%
C1 recall	8.65%	19.96%	20.33%	15.73%	31.08%
C1 F-measure	15.83%	32.44%	32.88%	26.75%	**44.61%**
C2 MAP	-	**46.01%**	44.75%	44.25%	44.41%
C3 BLEU	-	28.56%	-	**28.61%**	28.14%
German					
C1 precision	87.82%	77.78%	76.47%	81.70%	71.08%
C1 recall	8.54%	28.83%	30.49%	22.98%	38.92%
C1 F-measure	15.57%	42.07%	43.60%	35.87%	**50.30%**
C2 MAP	-	43.88%	**47.28%**	38.61%	46.42%
C3 BLEU	-	31.14%	-	**31.19%**	30.80%
Turkish					
C1 precision	93.16%	85.89%	85.43%	89.68%	79.38%
C1 recall	9.56%	19.53%	20.03%	17.78%	31.88%
C1 F-measure	17.35%	31.82%	32.45%	29.67%	**45.49%**

algorithm (Baseline) is compared to the Allomorfessor version presented in the Challenge 2008 (Alpha) [7], Morfessor Baseline [3], and Morfessor Categories-MAP (CatMAP) [4]. The first Morfessor Baseline, marked with an asterisk, was trained similarly to our Allomorfessor submission, whereas the second Morfessor Baseline and Morfessor CatMAP were trained with the full data sets.

From the results of Competition 1, we can note the following: The current Allomorfessor version clearly outperforms the old one, which tends to undersegment. Compared to Morfessor Baseline trained with the same data, the differences are small but statistically significant. For English, Allomorfessor has both higher recall and precision. For all the other languages, Morfessor has higher recall and lower precision, and thus also higher F-measure, as improving the lower measure (recall) affects the geometric mean more. Both Allomorfessor and Morfessor clearly benefit from the exclusion of rare word forms: Trained with the full data, Morfessor segments less and has higher precision, but much lower recall. Morfessor CatMAP outperforms both Allomorfessor and Morfessor Baseline in all the other languages of Competition 1 except for English, due to higher recall. In Competitions 2 and 3, Allomorfessor and Morfessor had no statistically significant differences. This implies that either the mutations are so rare that

they have no effect on the results, or the possible incorrect analyses cancel out the effect of the correct ones.

The amounts of mutations found by Allomorfessor are shown in Table 4. Generally, mutations are not used as much as linguistic analysis would prefer. The method finds, e.g., the morph `pretti` instead of deriving it as `pretty (y|i)`. The mutations in the English and Finnish results are analyzed in [10]. To summarize, a large part of the mutations correspond to a linguistic analysis. The most common error, especially for Finnish, is having a derived form as the base form. However, if the analyzed morph is semantically related to the induced base form, such analyses can be useful in applications. Other errors include not finding the linguistically correct suffix, using a more complex mutation and suffix combination than necessary, and using a semantically unrelated base form. Mutations are also used commonly on misspelled word forms. Overall, the problem seemed to be the low quantity of the mutations, not lack of quality.

The mutations are mostly used for morphs that occur only in few different word forms. In fact, this is a property of the Morfessor Baseline model: The model favors storing frequent words directly, as splitting them into parts decreases the likelihood significantly. In contrast, Morfessor CatMAP utilizes a *hierarchical* morph lexicon, where a word can be encoded by reusing the encodings of other words: E.g., if there are several word forms containing `uncertain`, the segmentation into `un` and `certain` needs to be stored only once and can be shared between all the different forms. This is beneficial especially for agglutinative languages, where words typically consist of many segments and, in part, explains the good results of CatMAP for them. The low recall of the previous Allomorfessor model [7], which also used a hierarchical lexicon, might then be due to other issues such as the assumed independence of mutations from morphemes.

Table 4. The number of non-empty mutations applied by Allomorfessor after the Viterbi analysis. Mutation usage is the number of non-empty mutation tokens divided by the number of morph tokens. For Arabic, "v" indicates vowelized script.

Language	Arabic	Arabic v	English	Finnish	German	Turkish
Mutation types	0	69	15	66	26	55
Mutation usage	0.0%	4.61%	0.18%	0.44%	0.17%	0.12%

5 Conclusions

We have described the Allomorfessor method for unsupervised morphological analysis. It attempts to find the morphemes of the input data by segmenting the words into morphs and finding mutations that can restore allomorphic variations in stems back to their base forms. In the Morpho Challenge 2009 evaluations, significant improvements were obtained over the previous version of the method. The results are now close to those of the Morfessor Baseline method. In comparison to the methods by the other participants, Allomorfessor performed especially well in the linguistic evaluation for English (the best result in the task) and in

the information retrieval and machine translation evaluations for all the languages. The main reason for not improving over Morfessor Baseline in most of the languages seems to be the low number of mutations applied by the algorithm.

Acknowledgments. This work was funded by Academy of Finland and Graduate School of Language Technology in Finland. We thank Jaakko Väyrynen and Mari-Sanna Paukkeri for comments on the manuscript, and Nokia and KAUTE foundations for financial support.

References

1. Bernhard, D.: Simple morpheme labelling in unsupervised morpheme analysis. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 873–880. Springer, Heidelberg (2008)
2. Bernhard, D.: MorphoNet: Exploring the use of community structure for unsupervised morpheme analysis. In: Working notes for the CLEF 2009 Workshop, Corfu, Greece (2009)
3. Creutz, M., Lagus, K.: Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Tech. Rep. A81, Publications in Computer and Information Science, Helsinki University of Technology (2005)
4. Creutz, M., Lagus, K.: Unsupervised models for morpheme segmentation and morphology learning. ACM Transactions on Speech and Language Processing 4(1) (2007)
5. Dasgupta, S., Ng, V.: High-performance, language-independent morphological segmentation. In: The Annual Conference of the North American Chapter of the ACL, NAACL-HLT (2007)
6. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Computational Linguistics 27(2), 153–189 (2001)
7. Kohonen, O., Virpioja, S., Klami, M.: Allomorfessor: Towards unsupervised morpheme analysis. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 975–982. Springer, Heidelberg (2009)
8. Kurimo, M., Virpioja, S., Turunen, V., Blackwood, G.W., Byrne, W.: Overview and results of Morpho Challenge 2009. In: Multilingual Information Access Evaluation 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30 - October 2. LNCS, vol. I, Springer, Heidelberg (2010)
9. Rissanen, J.: Stochastic Complexity in Statistical Inquiry, vol. 15. World Scientific Series in Computer Science, Singapore (1989)
10. Virpioja, S., Kohonen, O.: Unsupervised morpheme analysis with Allomorfessor. In: Working notes for the CLEF 2009 Workshop, Corfu, Greece (2009)
11. Yarowsky, D., Wicentowski, R.: Minimally supervised morphological analysis by multimodal alignment. In: Proceedings of the 38th Meeting of the ACL, pp. 207–216 (2000)

Unsupervised Morphological Analysis by Formal Analogy

Jean-François Lavallée and Philippe Langlais

DIRO, Université de Montréal
C.P. 6128, succursale Centre-ville
Montréal, Canada H3C 3J7
{lavalljf,felipe}@iro.umontreal.ca

Abstract. While classical approaches to unsupervised morphology acquisition often rely on metrics based on information theory for identifying morphemes, we describe a novel approach relying on the notion of *formal analogy*. A formal analogy is a relation between four forms, such as: `reader` is to `doer` as `reading` is to `doing`. Our assumption is that formal analogies identify pairs of morphologically related words. We first describe an approach which simply identifies all the formal analogies involving words in a lexicon. Despite its promising results, this approach is computationally too expensive. Therefore, we designed a more practical system which learns morphological structures using only a (small) subset of all formal analogies. We tested those two approaches on the five languages used in Morpho Challenge 2009.

1 Introduction

Two major approaches are typically investigated for accomplishing unsupervised morphological analysis. The first one uses statistics in order to identify the most likely segmentation of a word. The basic idea is that low predicability of the upcoming letter in a string indicates a morpheme boundary. This approach has been around for quite some time. Indeed, Harris [1] described such a system in the fifties. Variants of this idea have recently been investigated as well. For instance, both the system in [2] as well as *Morfessor* [3] utilize perplexity as one feature to score potential segmentations. The second approach consists of grouping words into paradigms and removing common affixes. Variants of this approach [4, 5] have yielded very good results in Morpho Challenge 2008 and 2009 [6].

The potential of *analogical learning* in solving a number of canonical problems in computational linguistics has been the subject of recent research [7–9]. In particular, several authors have shown that analogical learning can be used to accomplish morphological analysis. Stroppa & Yvon [10] demonstrate its usefulness in recovering a word's lemma. They report state-of-the-art results for three languages (English, Dutch and German). Hathout [11, 12] reports an approach where morphological families are automatically extracted thanks to formal analogies and some semantic resources. However, to the best of our knowledge, it has

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 617–624, 2010.

not been shown that analogical learning on a lexicon alone can be used as a means of acquiring a given language's morphology. This study aims to fill this gap.

The remainder of this paper is as follows. First, we provide our definition of formal analogy in Sect. 2. We then describe the two systems we devised based on this definition in Sect. 3. We present our experimental protocol and the results we obtained in Sect. 4. We conclude and discuss future avenues in Sect. 5.

2 Formal Analogy

A *proportional analogy*, or analogy for short, is a relation between four items noted $[x : y = z : t]$ which reads "x is to y as z is to t". Among proportional analogies, we distinguish formal analogies, that is, those we can identify at a graphemic level, such as [*cordially* : *cordial* = *appreciatively* : *appreciative*].

Formal analogies can be specified in many ways [13, 14]. In this study we define them in terms of factorization. Let x be a string over alphabet Σ, a *n-factorization* of x, noted f_x, is a sequence of n *factors* $f_x = (f_x^1, \ldots, f_x^n)$, such that $x = f_x^1 \odot f_x^2 \odot f_x^n$, where $\odot$ denotes the concatenation operator. Based on [14] we define a formal analogy as:

Definition 1. $\forall (x, y, z, t) \in {\Sigma^\star}^4$, $[x : y = z : t]$ ***iff*** *there exist d-factorizations* $(f_x, f_y, f_z, f_t) \in ({\Sigma^\star}^d)^4$ *of* (x, y, z, t) *such that:* $\forall i \in [1, d]$, $(f_y^i, f_z^i) \in \{(f_x^i, f_t^i), (f_t^i, f_x^i)\}$. *The smallest d for which this definition holds is called the degree of the analogy.*

According to this definition, [*cordially* : *cordial* = *appreciatively* : *appreciative*] is an analogy because we can find a quadruplet of 4-factorizations (factorizations involving 4 factors) as shown in the first column of Fig. 1. The second column of this figure also shows that a quadruplet of 2-factorizations also satisfies the definition. This illustrates the *alternations* passively captured by this analogy, that is, *appreciative*/*cordial* and *ly*/ϵ; the latter one (passively) capturing the fact that in English, an adverb can be constructed by appending *ly* to an adjective.

$f_{cordially}$	$\equiv$	*cordia*	*l l y*	$f_{cordially}$	$\equiv$	*cordial*	*ly*
$f_{cordial}$	$\equiv$	*cordia*	ϵ *l* ϵ	$f_{cordial}$	$\equiv$	*cordial*	ϵ
$f_{appreciatively}$	$\equiv$	*appreciative*	*l* ϵ *y*	$f_{appreciatively}$	$\equiv$	*appreciative*	*ly*
$f_{appreciative}$	$\equiv$	*appreciative*	ϵ ϵ ϵ	$f_{appreciative}$	$\equiv$	*appreciative*	ϵ

Fig. 1. Two factorizations of the analogy of degree 2 [*cordially* : *cordial* = *appreciatively* : *appreciative*]

3 Analogical Systems

The two systems we have designed rely on the assumption that a formal analogy implicitly identifies two pairs of forms that are morphologically related. For instance, the analogy in Fig. 1 relates *cordial* to *cordially*, as well as *appreciative* to *appreciatively*. Linking related words together is precisely the main task evaluated at Morpho Challenge. Therefore, given a lexicon $\mathcal{L}$, we need to identify all formal analogies involving its words. That is, we seek to compute:

$$\mathcal{A}(\mathcal{L}) = \{(x, y, z, t) \in \mathcal{L}^4 \ : \ [x : y = z : t]\} \tag{1}$$

Stroppa [15] describes a dynamic programming algorithm which checks whether a quadruplet of forms (x, y, z, t) is a formal analogy according to the previous definition. The complexity of this algorithm is in $O(|x| \times |y| \times |z| \times |t|)$.

As simple as it seems, identifying formal analogies is a very time consuming process. A straightforward implementation requires checking $O(|\mathcal{L}|^4)$ analogies, where $|\mathcal{L}|$ is the number of words in the lexicon. For all but tiny lexicons, this is simply not manageable. In order to accelerate the process, we used the *tree-count* strategy described in [8].

Unfortunately, computing $\mathcal{A}(\mathcal{L})$ for Morpho Challenge's largest lexicons still remains too time consuming.[1] Instead, we ran the analogical device on multiple languages for a week's time on randomly selected words. This enabled us to acquire a large set of analogies per language. From 11 (Arabic) to 52 (Turkish) million analogies were identified this way. While these figures may seem large at first, it is important to note that they represent only a fraction of the total potential analogies.

These sets of formal analogies are used by two systems we specifically designed for the first task of Morpho Challenge 2009. *Rali-Ana* is a pure analogical system, while *Rali-Cof* computes a set of `c-rules` (a notion we will describe shortly) which is used to accomplish the morphological analysis. The following sections describe both systems in detail.

3.1 Rali-Ana

This system makes direct use of the analogies we collected. Each time a word is involved in an analogy, we compute its factorization, as explained in Sect. 2. It is therefore possible to maintain a distribution over the *segmentations* computed for this word. The most frequent segmentation observed is kept by the system. Figure 2 illustrates the six segmentations observed for the 21 analogies involving the English word *abolishing* from which *Rali-Ana* selects *abolish+ing*.

It is important to note that because we computed only a small portion of all analogies, there are many words that this system cannot process adequately. In

[1] We roughly estimated that a few months of computation would be required for a single desk-computer to acquire all the possible analogies involving words in the Finnish lexicon of Morpho Challenge 2009.

particular, words for which no analogy is identified are added without modification to the final solution, clearly impacting recall.

abolish ing 12	*ab olishing* 4	*abol ishing* 2
a bo lishing 1	*abolis hing* 1	*abolish in g* 1

Fig. 2. Factorizations induced by analogy for the word *abolishing*. Numbers indicate the frequency of a given factorization.

3.2 Rali-Cof System

One drawback of *Rali-Ana* is that formal analogies capture information which is latent and highly lexical. For instance, knowing that [*cordial* : *cordially* = *appreciative* : *appreciatively*] does not tell us anything about [*cordial* : *appreciative* = *cordialness* : *appreciativeness*] or [*live* : *lively* = *massive* : *massively*]. Therefore, we introduce the notion of `c-rule` as a way to generalize the information captured by an analogy. Those `c-rules` are used by *Rali-Cof* in order to cluster together morphologically related words, thanks to a graph-based algorithm described hereafter.

CoFactor and C-Rule. In [8], the authors introduce the notion of *cofactor* of a formal analogy $[x : y = z : t]$ as a vector of d alternations $[\langle f, g\rangle_i]_{i\in[1,d]}$ where d is the degree (see Definition 1) of the analogy and an alternation is defined formally as:

$$\langle f, g\rangle_i = \begin{cases} (f_x^{(i)}, f_z^{(i)}) \text{ if } f_x^{(i)} \equiv f_y^{(i)} \\ (f_y^{(i)}, f_z^{(i)}) \text{ otherwise} \end{cases} \quad (2)$$

For instance, the cofactors for our running example are: [(*cordial*,*appreciative*), (ϵ,*ly*)]. Note that the pairs of factors in this definition are not directed, that is, (ϵ, *ly*) equals (*ly*, ϵ). Cofactors such as (ϵ, *ly*) or (*ity*, *ive*) represent suffixation operations frequently involved in English. Similarly, a cofactor such as (*un*, ϵ) which might capture a prefixation operation in English (*e.g.* *loved*/*unloved*) can relate a form such as *aunt* to the form *at*, just because the former happens to contain the substring *un*. Clearly, the generalization offered by a cofactor might introduce some noise if applied blindly.

This is the motivation behind the `c-rule`, a concept we introduce in this work. A `c-rule` is a directed cofactor which is expressed as a rewriting rule $\langle \alpha \rightarrow \beta \rangle$, where α and β are the two factors of a cofactor, such that $|\alpha| \geq |\beta|$.[2] As a result, applying a `c-rule` to a word always produces a shorter one.

In order to distinguish prefixation and suffixation operations which are very frequent, we add the symbol $\star$ to the left and/or to the right of the factors in order to indicate the existence of a non empty factors. In our running example, the two `c-rules` $\langle \star ly \rightarrow \star\epsilon \rangle$ and $\langle appreciative\star \rightarrow cordial\star \rangle$ are collected.

[2] In case both factors have the same length, alphabetical ordering is used.

For this paper, we note $\mathcal{R}(\mathtt{x})$, the application of the `c-rule` $\mathcal{R}$ on a word `x`. For instance, if $\mathcal{R}$ is $\langle \star\mathtt{ly} \rightarrow \star\epsilon \rangle$, $\mathcal{R}(\mathtt{elderly})$ equals `elder`. By direct extension, we also note $[\mathcal{R}_1, \ldots, \mathcal{R}_n](\mathtt{x})$ the form[3] resulting from the application of n `c-rules`: $\mathcal{R}_n(\ldots \mathcal{R}_2(\mathcal{R}_1(\mathtt{x}))\ldots)$.

Extraction of C-Rules. From the set of computed analogies, we extract every `c-rule` and its frequency of occurrence. As previously stated, the number of analogies generated is huge and so is the number of `c-rules`. Therefore, we applied a filter which removes low-frequency ones.[4] Relying on counts favors `c-rules` which contain short factors. For instance in English, the `c-rule` $\langle \mathtt{anti}\text{-}\star \rightarrow \epsilon\star \rangle$ is seen 2 472 times, while $\langle \mathtt{ka}\star \rightarrow \epsilon\star \rangle$, which is likely fortuitous, is seen 13 839 times. To overcome this, we further score a `c-rule` $\mathcal{R}$ by its *productivity* $prod(\mathcal{R})$ defined as the ratio of the number of times its application leads to a valid form over the number of times it can be applied. Formally:

$$prod(\mathcal{R}) = \frac{|\{\mathtt{x} \in \mathcal{L} : \mathcal{R}(\mathtt{x}) \in \mathcal{L}\}|}{|\{\mathtt{x} \in \mathcal{L} : \mathcal{R}(\mathtt{x}) \neq \mathtt{x}\}|} \tag{3}$$

Using productivity, the `c-rule` $\langle \mathtt{anti}\text{-}\star \rightarrow \epsilon\star \rangle$ has a score of 0.9490 compared to 0.2472 for $\langle \mathtt{ka}\star \rightarrow \epsilon\star \rangle$.

Word Relation Trees (WRT) Construction. *Rali-Cof* builds a forest of WRTs, where each tree identifies morphologically related words. A WRT is a structure where the nodes are the lexicon's words. An edge between nodes n_a and n_b, noted $n_a \rightarrow n_b$, is labelled by a set of `c-rules` which transforms word n_a into word n_b. The construction of the WRT forest is a greedy process, which applies the three following steps until all words in the lexicon have been processed:

1. Pick untreated word n from the lexicon.[5]
2. Compute set $\mathcal{S}(n)$ which contains words that can be reached by applying any strictly positive number of `c-rules` to word n.
3. Add an edge from n to $b \equiv \text{argmax}_{w \in \mathcal{S}(n)}\, score(n, w)$, the word of $\mathcal{S}(n)$ which maximizes a score (described hereafter), provided this score is greater than a given threshold.[6]

While building $\mathcal{S}(n)$ during step 2, it is often the case that different paths from word n to word w exist, as illustrated in Fig. 3. Therefore, the score between two words is computed by summing the score of each path. In turn, the score of one path $[\mathcal{R}_1, \ldots, \mathcal{R}_m]$, where $[\mathcal{R}_1, \ldots, \mathcal{R}_m](n) \equiv w$, is computed as $\prod_{i=1}^{m} prod(\mathcal{R}_i)$. If w happens to be the word selected at step 3, the retained path becomes an edge in the WRT labelled by the sequence of `c-rules` leading word n to word w.

[3] For the sake of clarity, we omit the case where the application of a `c-rule` leads to several forms.

[4] C-rules occurring less than 20 times are removed.

[5] The order in which the words are considered is unimportant.

[6] Set to 0.35 in this study.

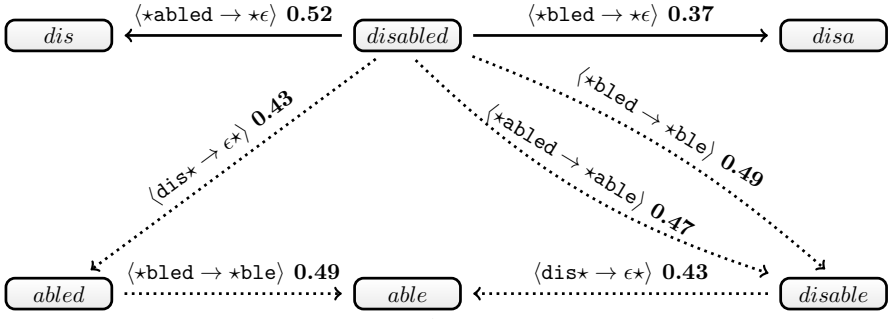


Fig. 3. Graph for the word *disabled*. The most probable link is *disable* with a score of 0.96. The dotted edges indicate the path considered for the computation of the score between *disabled* and *able*.

Segmentation into Morphemes. Each node in a WRT contains the segmentation of its associated word into its morphemes. In case of the root node, the set of morphemes is a singleton containing the word itself. For any other node (n), the set of morphemes is obtained by grouping together the morphemes of the father node (f) and those involved in the `c-rules` labeling the edge $n \rightarrow f$. To take one simple example, imagine a WRT contains the edge *disabled* $\rightarrow$ *able*, labelled by [$\langle\epsilon \rightarrow$ *dis*$\star\rangle$,$\langle\epsilon \rightarrow \star$ *d*$\rangle$]. The morphemes of *disabled* are [*able*,*dis*,*d*], where *dis* and *d* are the two morphemes present in the `c-rules`. As intuitive as it seems, the segmentation process is rather complex, the details of which are omitted for the sake of simplicity.

4 Experiments

The evaluation of the two systems we designed has been conducted by the Morpho Challenge 2009 organizers. The details of the evaluation protocol and the results can be found in [6]. Table 1 gives the official performance of our two systems compared to the one of *Morfessor* [3], a widely used system also employed as a baseline in Morpho Challenge. As can be observed, *Rali-Cof* outperforms both *Rali-Ana* and *Morfessor* for Finnish, Turkish and German.

The low recall of *Rali-Ana* can be explained by the fact that only a small subset of the analogies have been identified (See Sect.3.1). Nevertheless, the results yielded by this system are encouraging considering its simplicity. Especially since the precision for each language is rather good. We know that if we compute more analogies, recall will increase with the lexicon's coverage. Since the analyzed words were chosen without bias, precision will predictably not change much.

We observe that *Rali-Cof*'s performances are similar for all languages except for Arabic, for which we have a low recall. This might be caused by the provided lexicon's size, which is over 10 times inferior to that of the next smallest. Since analogical learning somehow relies on the pattern frequency to identify

morphemes, several valid morphemes might be overlooked due to their low frequency in the training set.

Although *Morfessor* has a higher F-Score in English, our approach surpasses it for languages with higher morphological complexity. This is noteworthy since the potential benefit of morphological analysis is greater for those languages.

Table 1. Precision (Pr.), Recall (Rc.) and F-measure (F1) for our systems and for the reference system, *Morfessor*, in the Morpho Challenge 2009 workshop

	Rali-Cof			*Rali-Ana*			*Morfessor Baseline*		
	Pr.	Rc.	F1	Pr.	Rc.	F1	Pr.	Rc.	F1
ENG.	68.32	46.45	55.30	64.61	33.48	44.10	**74.93**	**49.81**	**59.84**
FIN.	**74.76**	**26.20**	**38.81**	60.06	10.33	17.63	89.41	15.73	26.75
TUR.	**48.43**	**44.54**	**46.40**	69.52	12.85	21.69	89.68	17.78	29.67
GER.	**67.53**	**34.38**	**45.57**	61.39	15.34	24.55	81.70	22.98	35.87
ARB.	94.56	2.13	4.18	92.40	4.40	8.41	**91.77**	**6.44**	**12.03**

5 Discussion and Future Work

We have presented the two systems we designed for our participation in Morpho Challenge 2009. While both use formal analogy, *Rali-Cof* extracts the lexicalized information captured by an analogy through the use of `c-rules` a concept we introduced here. While *Rali-Ana* requires computing the full set of analogies involving the words found in a lexicon, *Rali-Cof* only requires a (small) subset of those analogies to function correctly and is therefore more practical.

Considering that only a fraction of the total available words have been processed by *Rali-Ana*, its performances are rather promising. We are also pleased to note that *Rali-Cof* outperforms a fair baseline (*Morfessor*) on Turkish, Finnish and German.

We developped our systems within a very short period of time, making many hard decisions that we did not have time to investigate further. This reinforces our belief that formal analogies represent a principled concept that can be efficiently used for unsupervised morphology acquisition.

Still, a number of avenues remain to be investigated. First, we did not adjust the meta-parameters controlling the *Rali-Cof* system to a specific language. This could be done using a small supervision set, that is, a set of words that are known to be morphologically related. Second, we plan to investigate the impact of the quantity of analogies computed. Preliminary experiments showed that in English, formal analogies computed on less than 10% of the words in the lexicon could identify most of the major affixes. Third, while `c-rules` capture more context than cofactors do, other alternatives might be considered, such as regular expressions, as in [16]. Last, we observed that sometimes, words in a WRT are not morphologically related. We think it is possible to consider formal analogies in order to filter out some associations made while constructing the WRT forest.

References

1. Harris, Z.S.: From phoneme to morpheme. Language 31, 190–222 (1955)
2. Bernhard, D.: Simple morpheme labelling in unsupervised morpheme analysis. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 873–880. Springer, Heidelberg (2008)
3. Creutz, M., Lagus, K.: Inducing the morphological lexicon of a natural language from unannotated text. In: Proceedings of AKRR 2005, vol. 5, pp. 106–113 (2005)
4. Monson, C., Carbonell, J., Lavie, A., Levin, L.: Paramor: Minimally supervised induction of paradigm structure and morphological analysis. In: Proceedings of 9th SIGMORPHON Workshop, Prague, Czech Republic, pp. 117–125. ACL (June 2007)
5. Zeman, D.: Using unsupervised paradigm acquisition for prefixes. In: CLEF 2008 Workshop, Arhus, Denmark (2008)
6. Kurimo, M., Virpioja, S., Turunen, V., Blackwood, G., Byrne, W.: Overview and results of morpho challenge 2009. In: 10th CLEF Workshop, Corfu, Greece (2010)
7. Lepage, Y., Denoual, E.: Purest ever example-based machine translation: Detailed presentation and assessment. Machine Translation 29, 251–282 (2005)
8. Langlais, P., Patry, A.: Translating unknown words by analogical learning. In: EMNLP-CoNLL, Prague, Czech Republic, pp. 877–886 (2007)
9. Denoual, E.: Analogical translation of unknown words in a statistical machine translation framework. In: MT Summit, XI, Copenhagen September 10-14 (2007)
10. Stroppa, N., Yvon, F.: An analogical learner for morphological analysis. In: CoNLL, Ann Arbor, MI, pp. 120–127 (June 2005)
11. Hathout, N.: From wordnet to celex: acquiring morphological links from dictionaries of synonyms. In: 3rd LREC, Las Palmas de Gran Canaria, pp. 1478–1484 (2002)
12. Hathout, N.: Acquistion of the morphological structure of the lexicon based on lexical similarity and formal analogy. In: 3rd Textgraphs Workshop, Manchester, United Kingdom, pp. 1–8 (August 2008)
13. Pirrelli, V., Yvon, F.: The hidden dimension: a paradigmatic view of data-driven NLP. Journal of Experimental & Theroretical Artifical Intelligence 11, 391–408 (1999)
14. Yvon, F., Stroppa, N., Delhay, A., Miclet, L.: Solving analogical equations on words. Technical Report D005, ENST, Paris, France (July 2004)
15. Stroppa, N.: Définitions et caractérisations de modèles à base d'analogies pour l'apprentissage automatique des langues naturelles. PhD thesis, ENST, ParisTech, Télécom, Paris, France (November 2005)
16. Bernhard, D.: Morphonet: Exploring the use of community structure for unsupervised morpheme analysis. In: 10th CLEF Workshop, Corfu, Greece (2010)

Unsupervised Word Decomposition with the Promodes Algorithm

Sebastian Spiegler, Bruno Golénia, and Peter A. Flach

Machine Learning Group, Computer Science Department, University of Bristol, UK
{spiegler,goleniab,flach}@cs.bris.ac.uk

Abstract. We present Promodes, an algorithm for unsupervised word decomposition, which is based on a probabilistic generative model. The model considers segment boundaries as hidden variables and includes probabilities for letter transitions within segments. For the Morpho Challenge 2009, we demonstrate three versions of Promodes. The first one uses a simple segmentation algorithm on a subset of the data and applies *maximum likelihood estimates* for model parameters when decomposing words of the original language data. The second version estimates its parameters through *expectation maximization* (EM). A third method is a *committee of unsupervised learners* where learners correspond to different EM initializations. The solution is found by majority vote which decides whether to segment at a word position or not. In this paper, we describe the probabilistic model, parameter estimation and how the most likely decomposition of an input word is found. We have tested Promodes on non-vowelized and vowelized Arabic as well as on English, Finnish, German and Turkish. All three methods achieved competitive results.

1 Introduction

Morphological analysis can be defined as study of the internal word structure [1]. According to [2], there are four tasks involved in morphological analysis: 1) decomposing words into morphemes, 2) building a morpheme dictionary, 3) defining morphosyntactical rules stating how morphemes are combined to valid words and 4) defining morphophonological rules specifying phonological changes when combining morphemes. For the Morpho Challenge, the task is *unsupervised morpheme analysis* of words contained in a word list using a generic algorithm without any further information.

Related Work. Goldsmith [3] presented the morphological analyzer *Linguistica* which learns *signatures*. A similar approach has been chosen by Monson [4] who developed *Paramor*, an algorithm which induces *paradigmatic structures* of morphology. *Morfessor* is a model family for unsupervised morphology induction developed by Creutz et al. [5]. The two main members of this family are *Morfessor baseline* based on minimum description length (MDL) and *Morfessor Categories-MAP* with a probabilistic maximum a posteriori (MAP) framework and mor-

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 625–632, 2010.

pheme categories.[1] Linguistica, Paramor and Morfessor carry out morphological analysis in terms of word decomposition, learning a morpheme dictionary and finding morphosyntactical rules. Other approaches [6,7] focused on word decomposition by analyzing words based on *transition probabilities* or *letter successor variety* which originates in Harris' approach [8]. Moreover, Snover [9] described a *generative model* for unsupervised learning of morphology, however, it differs from ours. Snover searched, similar to Monson, for paradigms, whereas we are interested in word decomposition based on the probability of having a boundary in a certain position and the resulting letter transition of morphemes. The remainder of the paper is structured as follows. In Sec. 2 we present PROMODES, its mathematical model, the parameter estimation and word decomposition. In Sec. 3 and 4 experiments are explained, results analysed and conclusions drawn.

2 Algorithm

The PROMODES algorithm is based on a probabilistic generative model which can be used for word decomposition when fully parameterized. Its parameters can be estimated using *expectation maximization (EM)* [10] or by computing *maximum likelihood estimates (MLE)* from a pre-segmented training set. Independently of the parameter estimation, either a single model is used for decomposition or a set of separate models as a *committee of unsupervised learners* where η different results are combined by majority vote. In Sec. 2.1 we will introduce the PGM and show how we apply it to find a word's best segmentation. Subsequently, in Sec. 2.2 we will explain how we estimate model parameters and in Sec. 2.3 we demonstrate how a committee of unsupervised learners is used to decompose words.

2.1 Probabilistic Generative Model

A *probabilistic generative model (PGM)* is used to describe the process of data generation based on observed variables X and target variables Y with the goal of forming a conditional probability distribution $Pr(Y|X)$. In morphological analysis the observables correspond to the original words and the hidden variables to their segmentations. A word w_j from a list W with $1 \leq j \leq |W|$ consists of n letters and has $m = n-1$ positions for inserting boundaries. A word's segmentation b_j is described by a binary vector $(b_{j1}, \ldots, b_{jm})$. A boundary value b_{ji} is $\{0, 1\}$ depending on whether a boundary is inserted or not in i with $1 \leq i \leq m$. A letter transition t_{ji} consists of letter $l_{j,i-1}$ and l_{ji}, which belong to some alphabet, and traverses position i in w_j. By convention, l_{j0} is the first letter of w_j.

Finding a Word's Segmentation. Since a word has an exponential number of possible segmentations[2], it is prohibitive to evaluate all of them in order to find the most likely one. Therefore, the observables in our model are letter

[1] Both morphological analyzers are reference algorithms for the Morpho Challenge.
[2] A word can be segmented in 2^m different ways with $m = n-1$ and n as letter length.

transitions t_{ji} with $Pr(t_{ji}|b_{ji})$ and the hidden variables are the boundary values b_{ji} with $Pr(b_{ji})$ assuming that a boundary in i is inserted independently of other positions. Knowing the parameters of the model, the *letter transition probability distribution* and the *probability distribution over non-/boundaries*, we can find the best segmentation of a given word with $2m$ evaluations using (1).

$$\arg\max_{b_{ji}} Pr(b_{ji}|t_{ji}) = \begin{cases} 1, & \text{if } Pr(b_{ji}=1)\cdot Pr(t_{ji}|b_{ji}=1) \\ & > Pr(b_{ji}=0)\cdot Pr(t_{ji}|b_{ji}=0) \\ 0, & \text{otherwise .} \end{cases} \quad (1)$$

Below, we will define the two parameter distributions explicitly.

Letter Transition Probability Distribution. In the Markovian spirit we describe a word by transitions from letters x to y within a morpheme where y is drawn from alphabet A and x from $A_{\mathcal{B}} = A \cup \{\mathcal{B}\}$ where $\mathcal{B}$ is a silent start symbol pointing to the first letter of a morpheme. By introducing such a symbol it is guaranteed that all segmentations of a word have the same number of transitions.

$$p_{x,y} = Pr(X_i = y | X_{i-1} = x) \quad (2)$$
$$\text{with } \sum_{y \in A} p_{x,y} = 1 \ \forall x \in A_{\mathcal{B}} \text{ and } 1 \leq i \leq m \ .$$

Equation (2) is used in (7) and (8) for describing the probability of a letter transition in position i in the PGM.

Probability Distribution over Non-/Boundaries For describing a segmentation we chose a *position-dependent and non-identical distribution*. Each position i is therefore assigned to a Bernoulli random variable Z_i and the existence of a boundary corresponds to a single trial with positive outcome.

$$p_{z_i,m} = Pr(Z_i = 1|m) \quad (3)$$

with $Pr(Z_i = 0|m) + Pr(Z_i = 1|m) = 1$, $1 \leq i \leq m$ and $Z_i \in Z$. The model can be summarised as $\theta = \{X, Z\}$. Equation (3) is subsequently applied to define the probability of segmenting in position i.

Probability of Segmenting in Position i Derived from (3) the probability of putting a boundary in position i is defined as

$$Pr(b_{ji}|m, \theta) = p_{z_i,m} \quad (4)$$

where $p_{z_i,m}$ is the probability of having a boundary value $b_{ji} = z_i$ in i given length m of the segmentation. We rewrite this equation as

$$Pr(b_{ji}|m, \theta) = \prod_{r=0}^{1} (p_{r,m})^{\mu_{b_{ji},r,i,m}} \ , \quad (5)$$

$$\mu_{b_{ji},r,i,m} = \begin{cases} 1, & \text{if } b_{ji} = r \text{ in position } i \text{ given } m \ , \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where we iterate over possible boundary values $r = \{0, 1\}$ and use $\mu_{b_{ji},r,i,m}$ to eliminate all r's in the product which do not correspond to b_{ji} in i given m.

Probability of a Letter Transition in Position i. If the segmentation in i is known we can assign a letter transition probability based on (2) and get

$$Pr(t_{ji}|b_{ji}, \theta) = p_{x,y} \tag{7}$$

where transition t_{ji} consists of letter $l_{j,i-1} = x$ and $l_{ji} = y$ given boundary value b_{ji} in position i. For later derivations, we rewrite (7) such that we iterate over the alphabet using x' and y', and eliminate all probabilities which do not correspond to the original x and y using function $\mu_{xy,x'y'}$.

$$Pr(t_{ji}|b_{ji}, \theta) = \prod_{x',y' \in A} (p_{x,y})^{\mu_{xy,x'y'}} \quad . \tag{8}$$

$$\mu_{xy,x'y'} = \begin{cases} 1, & \text{if } x' = x \text{ and } y' = y \ , \\ 0, & \text{otherwise} \ . \end{cases} \tag{9}$$

Finding the Best Segmentation of a Word. With (5) and (8) the solution of the problem in (1) becomes

$$b^*_{ji} = \begin{cases} 1, & \text{if } Pr(Z_i = 1|m_j) \cdot Pr(X_i = l_i|X_{i-1} = \mathcal{B}) \\ & > Pr(Z_i = 0|m_j) \cdot Pr(X_i = l_i|X_{i-1} = l_{i-1}) \ , \\ 0, & \text{otherwise} \ , \end{cases} \tag{10}$$

$$b^*_j = (b^*_{j,1}, \ldots, b^*_{j,m}) \ . \tag{11}$$

2.2 Parameter Estimation

Before applying the probabilistic model, its parameters have to be estimated using maximum likelihood estimates or expectation maximization.

Maximum Likelihood Estimates (MLE). We segmented each training set using a heuristic similar to the *successor variety* [8] in a *separate* pre-processing step. All possible substrings of each word were collected in a forward trie along with statistical information, e.g. frequencies. A particular word was decomposed based on probabilities of a letter following a certain substring. From the segmentations we estimated the parameters of Promodes 1 (P1) using MLE.

Expectation Maximization (EM). Parameter estimation by EM [10] was used in Promodes 2 (P2). The EM algorithm iteratively alternates between two distinctive steps, the expectation or E-step and the maximization or M-step, until a convergence criterion is met. In the E-step the log-likelihood of the current estimates for the model parameters are computed. In the M-step the parameters

are updated such that the log-likelihood is maximized. The Q function as the expected value of the log-likelihood function is defined as:

$$Q(\theta, \theta_t) = \sum_{j=1}^{|W|} \sum_{i=1}^{m_j} \sum_{r=0}^{1} \left(Pr(b_{ji} = r | t_{ji}, \theta_t) \log Pr(t_{ji}, b_{ji} = r | \theta) \right) \quad , \tag{12}$$

$$\theta^* = \arg\max_{\theta} Q(\theta, \theta_t) \quad . \tag{13}$$

The objective function which we want to maximize during the *M-step* is built from the Q function and includes constraints and Lagrange multipliers.[3] The parameters of the model are re-estimated by using partial derivatives which result in the new estimates for the letter transition probabilities as

$$\hat{p}_{x,y} = \frac{\sum_{j=1}^{|W|} \sum_{i=1}^{m_j} \sum_{r=0}^{1} \left(Pr(b_{ji} = r | t_{ji}, \theta_t) \sum_{x',y' \in A} \mu_{xy,x'y'} \right)}{\sum_{y' \in A} \sum_{j=1}^{|W|} \sum_{i=1}^{m_j} \sum_{r=0}^{1} \left(Pr(b_{ji} = r | t_{ji}, \theta_t) \sum_{x'',y'' \in A} \mu_{xy',x''y''} \right)} \quad , \tag{14}$$

and for the probability distribution over boundary positions as

$$\hat{p}_{z_i,m} = \frac{\sum_{j=1}^{|W|} \sum_{i=1}^{m_j} \sum_{r=0}^{1} \left(Pr(b_{ji} = r | t_{ji}, \theta_t) \sum_{r'=0}^{1} \mu_{z_i,r',i,m} \right)}{\sum_{r'=0}^{1} \sum_{j=1}^{|W|} \sum_{i=1}^{m_j} \sum_{r=0}^{1} \left(Pr(b_{ji} = r | t_{ji}, \theta_t) \sum_{r''=0}^{1} \mu_{r',r'',i,m} \right)} \quad . \tag{15}$$

Although both estimates look complicated, they have an intuitive interpretation. In (14) we count the occurrences of letter transitions from x to y weighted by the posterior probability $Pr(b_{ir}|t_{ji}, \theta_t)$ and divide it by the weighted sum of all transitions from x. In (15) the weighted sum for putting a boundary in i of words with length m is divided by the weighted sum of all boundaries and non-boundaries in i for the same words.

2.3 Committee of Unsupervised Learners

Since different initializations of the EM may converge in different local optima, corresponding models might give slightly different analyses for a single word. Promodes Committee (PC) averages results varying initializations and combines them into a single solution using a *committee of unsupervised learners* similar to [12]. A committee can combine results from different algorithms or in our case different initializations. Each committee member can vote for a certain partial or complete solution. The weight of each vote can be uniform or non-uniform, e.g. based on performance or confidence of the algorithm. Our approach is completely unsupervised and purely based on *majority vote* where each vote for putting a boundary in a certain position counts equally. Given η analyses for a single word w_j in position i we introduce $score_{j,i}$ as

[3] The objective function is specified in detail in [11].

$$score_{j,i} = \sum_{h=1}^{\eta} \pi_{h,j,i} \tag{16}$$

$$\pi_{h,j,i} = \begin{cases} +1, \text{ if analysis } h \text{ contains boundary in } i \text{ of word } w_j, \\ -1, \text{ otherwise} \end{cases} \tag{17}$$

and put a boundary at the ith position of word w_j if $score_{j,i} > 0$.

3 Experimental Results

Although PROMODES is intended for agglutinating languages like Finnish and Turkish, it was also applied to fusional languages like Arabic, German and English. PROMODES decomposes words into their morphemes. Morphosyntactic rules are implicitly stored as statistics in the form of probabilities for segmenting at certain word positions and probabilities for the resulting letter transitions within morphemes. There is no further grammatical analysis like building *signatures* or *paradigms*. Morpheme labels are the morphemes themselves or simple labels consisting of *morpheme[index number]*. The results across languages are listed in Tab. 1 with the highest precision, recall and f-measure written in bold.

General Setup of Experiments. Independently of the PROMODES version, we generated a training subset for each language consisting of 100,000 words randomly sampled[4] from each corpus. In the case of Arabic, we employed the entire corpus since it contained less words. For P1 we estimated parameters from the pre-segmented training set which was generated with a simple segmentation algorithm described in Sec. 2.2. By using MLE we averaged statistics across the subset. Subsequently, the model was applied to the entire dataset to decompose all words. P2 used EM to estimate its parameters. Initially, words from the training set were randomly segmented and then the EM algorithm improved the parameter estimates until the convergence criterion was met.[5] The resulting probabilistic model was then applied to the entire dataset. PC made use of the different initializations and resulting analyses of P2. Instead of choosing a single result a committee, described in Sec. 2.3, combined different solutions into one.

Analysis of Results. In general, PROMODES performed best on non-vowelized (nv.) and vowelized (vw.) Arabic, well on Finnish and Turkish, and moderately on English and German compared to other approaches in the Morpho Challenge 2009. For a detailed comparison see [14]. Of the three PROMODES versions there was no best method for all languages. An analysis of the different gold standards suggested, however, that all PROMODES methods performed better on languages with a high morpheme per word ratio. In detail, the best result

[4] No frequency or word length considerations.

[5] We used the *Kullback-Leibler divergence* [13] which measures the difference between the model's probability distributions before and after each iteration of the EM.

(f-measure) for English was achieved by P1, for Arabic (nv./vw.), German and Turkish by P2, and for Finnish by PC. Especially for nv. Arabic, Promodes achieved a high precision (at the cost of a lower recall). This implies that most morphemes returned were correct but only few were found. The reason for that might be that words were quite short (5.77 letters on av.)[6] and lacking the grammatical information carried by the vowels. Furthermore, words contained more morpheme labels per word (8.80 morphemes on av.) than letters which made morpheme analysis challenging. Promodes showed better results on vw. Arabic which were also more balanced between precision and recall. Especially for English with longer words (8.70 letters on av.) and fewer morphemes per word (2.25 morphemes on av.), Promodes exhibited a different behavior with a low precision but a high recall. This suggests that the algorithm splits words into too many morphemes. A similar effect was encountered for the morphologically more complex languages Finnish and Turkish where Promodes tended to oversegment as well. For German, precision and recall varied a lot with different Promodes versions so that a general pattern could not be identified.

Table 1. Results of P1, P2, PC in Competition 1

Language	Precision			Recall			F-measure		
	P1	P2	PC	P1	P2	PC	P1	P2	PC
Arabic (nv)	**.8110**	.7696	.7706	.2057	**.3702**	.3696	.3282	**.5000**	.4996
Arabic (vw)	**.7485**	.6300	.6832	.3500	**.5907**	.4797	.4770	**.6097**	.5636
English	**.3620**	.3224	.3224	**.6481**	.6110	.6110	**.4646**	.4221	.4221
Finnish	.3586	.3351	**.4120**	.5141	**.6132**	.4822	.4225	.4334	**.4444**
German	**.4988**	.3611	.4848	.3395	**.5052**	.3461	.4040	**.4212**	.4039
Turkish	.3222	.3536	**.5530**	**.6642**	.5870	.2835	.4339	**.4414**	.3748

4 Conclusions

We have presented three versions of the Promodes algorithm which is based on a probabilistic model. The parameters of Promodes 1 (P1) were estimated using maximum likelihood estimates. Expectation maximization was applied in Promodes 2 (P2). Promodes Committee (PC) combined results from different initialisations of P2 by using a committee of unsupervised learners. All three methods achieved competitive results in the Morpho Challenge 2009. The strengths of Promodes, in general, are that it does not make assumptions about the structure of the language in terms of prefix and suffix usage. Furthermore, instead of building a morphological dictionary and a rule base which are likely to be incomplete, it applies statistics of a small training set to a larger test set. This is achieved at the cost of over-segmenting since there is no inductive bias towards a compressed morphological dictionary. Our future work includes

[6] Average measures based on the respective gold standard.

extending the probabilistic model to a higher order which should increase the model's memory and therefore reduce over-segmentation. We also intend to further analyse the behaviour of the committee and examine the impact of different training set sizes.

Acknowledgement

We would like to thank Aram Harrow for fruitful discussions on the mathematical background of this paper, our team colleagues Roger Tucker and Ksenia Shalonova for advising us on general issues in morphological analysis and the anonymous reviewers for their comments. This work was sponsored by EPSRC grant EP/E010857/1 *Learning the morphology of complex synthetic languages*.

References

1. Booij, G.: The Grammar of Words: An Introduction to Linguistic Morphology. Oxford University Press, Oxford (2004)
2. Goldsmith, J.: Segmentation and Morphology. The Handbook of Computational Linguistics. Blackwell, Malden (2009)
3. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Computational Linguistics 27 (2001)
4. Monson, C.: ParaMor: From Paradigm Structure To Natural Language Morphology Induction. PhD thesis, Carnegie Mellon University, Pittsburgh, PA (2008)
5. Creutz, M., Lagus, K.: Inducing the morphological lexicon of a natural language from unannotated text. In: Proc. of AKRR, vol. 1 (2005)
6. Bernhard, D.: Simple morpheme labeling in unsupervised morpheme analysis. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 873–880. Springer, Heidelberg (2008)
7. Dang, M.T., Choudri, S.: Simple unsupervised morphology analysis algorithm (sumaa). In: Proc. of PASCAL Workshop on Unsuperv. Segmentation of Words into Morphemes, Italy, vol. 1 (2006)
8. Harris, Z.S.: From phoneme to morpheme. Language 31 (1955)
9. Snover, M.G., Brent, M.R.: A probabilistic model for learning concatenative morphology. In: Proc. of NIPS (2002)
10. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithms. Journal of the Royal Statistical Society 39 (1979)
11. Spiegler, S., Golenia, B., Flach, P.: Promodes: A probabilistic generative model for word decomposition. In: Working Notes for the CLEF 2009 Workshop, Greece (2009)
12. Atwell, E., Roberts, A.: Combinatory hybrid elementary analysis of text (cheat). In: Proc. of PASCAL Workshop on Unsuperv. Segmentation of Words into Morphemes, Italy, vol. I (2006)
13. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics 22 (1951)
14. Kurimo, M., Virpioja, S., Turunen, V.T., Blackwood, G.W., Byrne, W.: Overview of morpho challenge 2009. In: Multilingual Information Access Evaluation, CLEF 2009, Greece. LNCS, vol. I. Springer, Heidelberg (2010)

Unsupervised Morpheme Discovery with Ungrade

Bruno Golénia, Sebastian Spiegler, and Peter A. Flach

Machine Learning Group, Computer Science Department, University of Bristol, UK
{goleniab,spiegler,flach}@cs.bris.ac.uk

Abstract. In this paper, we present an unsupervised algorithm for morpheme discovery called UNGRADE (UNsupervised GRAph DEcomposition). UNGRADE works in three steps and can be applied to languages whose words have the structure prefixes-stem-suffixes. In the first step, a stem is obtained for each word using a sliding window, such that the description length of the window is minimised. In the next step prefix and suffix sequences are sought using a morpheme graph. The last step consists in combining morphemes found in the previous steps. UNGRADE has been experimentally evaluated on 5 languages (English, German, Finnish, Turkish and Arabic) with encouraging results.

1 Introduction

Morphological analysis is concerned in part with the process of segmenting words of a given corpus into their morphemes. Morphemes are the smallest units in a word bearing a meaning. In various applications such as speech synthesis, we need a language model which should describe all possible words, preferably in a complete manner. One way of obtaining such a model is by simply listing words in a dictionary. This, however, is not feasible for most languages since there exists a large number of possible words. Furthermore, for a given corpus, the quantity of different morphemes is usually smaller than the quantity of different words. From this follows that it is more meaningful to create a dictionary of morphemes using morphological analysis. Analysis can be either carried out manually by linguistic experts or in an automated fashion by a machine. Since manual analysis is time-consuming and labour-intensive it is worth studying machine learning approaches to reduce the quantity of work needed to create a vocabulary of morphemes by linguistic experts.

In the past, research in computational linguistics mainly focused on unsupervised morphological analysis for large datasets with approaches like Linguistica [9] and Morfessor [4]. Shalonova et al. introduced a semi-supervised approach specifically designed for small data sets [17]. Spiegler et al. compared the approach with Morfessor [18]. In this paper, we extend the semi-supervised approach of Shalonova et al. for large data sets and make it unsupervised through window-based pre-processing. We assume that each word follows the structure prefixes, a stem and suffixes without limiting on the number of prefixes and suffixes.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 633–640, 2010.

The method that we propose can be broken down into three steps. First of all, stems are found using a window under a MDL (Minimum Description Length) paradigm. Secondly, a graph-based algorithm GBUMS (Graph-Based Unsupervised Morpheme Segmentation) is applied to determine independently the sequences of prefixes and suffixes. GBUMS follows a bottom-up approach, where each node represents a morpheme. GBUMS merges morphemes according to a lift function and uses a stopping criterion based on BIC (Bayesian Information Criterion) to stop merging. Finally, in the last step, UNGRADE aggregates the segmentations from the previous steps.

2 Related Work

Much research on *Unsupervised segmentation of morphology* has focused on statistical approaches according to the work of Harris [12,6] and tuning of parameters according to a language like Gaussier [7]. Brent (1993) presented the MDL theory for Computational linguistic problems with a probabilistic approach using the spelling of words [1]. Afterwards, Brent (1995) defined an approach for finding suffixes in a language [2]. Unfortunately, Brent required a special tagging of the data. Subsequently, Goldsmith (2001) used MDL [9] to combine the results of multiple heuristics based on statistic like [6,7] in a software called *Linguistica*. Goldsmith defined a model for MDL with *signature*. However, Linguistica was only focused on European languages. More recently, Creutz et al. [4] (2005) presented *Morfessor* with two new approaches to discover morphemes named *Morfessor baseline* and *Morfessor Categories-MAP*. The former method was based on MDL in a recursive method. The latter one, the most efficient, combined Maximum A Posteriori and Viterbi for an optimal segmentation. Morfessor was developed independently of languages and provided good results. Lately, Paramor (2007) developed by Monson, in a similar way to Goldsmith with *signatures*, used *paradigm* without using MDL [15]. Paramor worked in two steps and provided results as good as Morfessor. In 2008, results from Paramor and Morfessor were combined and provided better results than one of them alone [16].

3 Stem Extraction Using a Window of Letters

Extraction of morpheme sequences is a hard task in languages where the word form includes a sequence of prefixes, a stem and sequence of suffixes. However, by finding the boundaries of the stem first, it is possible to extract prefix sequences and suffix sequences efficiently. In order to extract the stems, we look for a stem in the middle of each word, assuming it as the most often position overlapping the real stem. We develop a heuristic to seek the most probable stem given a word through a window by using the MDL principle. We define a window by two boundaries within a word between letters. In other words, a window is a substring of a word. During initialisation, the width is fixed to 1 and the window corresponds to a single letter which is the middle of the word. Thereafter, an algorithm is used to shift, increase or decrease the width of the window from

its initial point to its left and/or to its right side. Consequently, an evaluation function is used for each window and repeated for the best windows until no better windows are found. The final boundaries are considered as the limit of the stem in the word. We iterate the algorithm for each word in the corpus, the evaluation function used is the *Minimum Description Length Window Score* (MDLWS).

Definition 1. *Let* $win = (l_{win}, r_{win})$ *be a window with a left boundary* l_{win} *and a right boundary* r_{win}*. Given a word* w *and a window* win*, the MDLWS is defined by:*

$$MDLWS(win, w) = log_2(r_{win} - l_{win} + 1) + log_2(np(w, l_{win}, r_{win})) \quad (1)$$

where np denotes the n-gram probability of the window win in the word w*.*

The n-gram probability is estimated using all n-subsequences of each word in a corpus. Therefore, the best window is acquired by the minimum of MDLWS using the successive application of three operators (shift, increase and decrease) from the initial point of the window. As soon as the algorithm has been applied to each word from a corpus, we process the left side of each window to extract the prefixes. In a similar way, we process the right side of the window to extract the suffixes. To do so, we use the GBUMS (Graph-Based Unsupervised Morpheme Segmentation) algorithm presented in the next section.

4 Graph-Based Unsupervised Morpheme Segmentation

In the previous section, we showed how to find stems. In this section, we introduce GBUMS (Graph-Based Unsupervised Morpheme Segmentation) to extract sequences of prefixes and suffixes. The algorithm GBUMS was developed in [10] under the name GBUSS (Graph-Based Unsupervised Suffix Segmentation) to extract suffix sequences efficiently and applied to Russian and Turkish languages on a training set in [10,17]. Afterwards, we use GBUSS to extract independently both prefix and suffix sequences, instead of only suffix sequences. We refer to prefixes and suffixes generally as morphemes. We call GBUMS the extended version of the GBUSS algorithm for morpheme extraction. We call L-corpus (R-corpus) the list of substrings obtained from the left-side (right-side) of the windows. In an independent manner, we use the term M-corpus for a L-corpus (R-corpus) in a prefix (suffix) graph-based context. GBUMS uses a morpheme graph in a bottom-up fashion. Similar to Harris [12], we base our algorithm on letter frequencies. However, where Harris builds on successor and predecessor frequencies, we use position-independent n-gram statistics to merge single letters to morphemes until a stopping criterion has been met. In the morpheme graph, each node represents a morpheme and each directed edge the concatenation of two morphemes labeled with the frequencies in a M-corpus (example on Figure 1).

Definition 2. *Let* $M = \{m_i | 1 \leq i \leq |M|\}$ *be a set of morphemes, let* f_i *be the frequency with which morpheme* m_i *occurs in a M-corpus* X *of morpheme sequences, let* $v_i = (m_i, f_i)$ *for* $1 \leq i \leq n$*, and let* $f_{i,j}$ *denote the number of morpheme sequences in the corpus in which morpheme* m_i *is followed by morpheme* m_j*. The* morpheme graph $G = (V, E)$ *is a directed graph with vertices or nodes* $V = \{v_i | 1 \leq i \leq |V|\}$ *and edges* $E = \{(v_i, v_j) | f_{i,j} > 0\}$*. We treat* $f_{i,j}$ *as the label of the edge from* v_i *to* v_j*.*

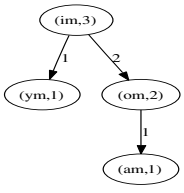


Fig. 1. Morpheme graph with 4 nodes having morphemes im, ym, om and am

In G, each node is initialised with a letter according to a M-corpus X, then, one by one nodes are merged to create the real morphemes. To merge nodes an evaluation function is necessary. In [17], Shalonova et al. proposed one based on frequency and entropy. For large data sets, due to high computational costs we simplify the equation and do not take the entropy into account. The approach that we present can be viewed as the *lift* of a rule for association rules in data mining [3]. Consequently, we name the evaluation function *Morph_Lift*.

Definition 3. *Morph_Lift is defined for a pair of morphemes* m_1 *and* m_2 *as follows:*

$$Morph\_Lift(m_1, m_2) = \frac{f_{1,2}}{f_1 + f_2} \tag{2}$$

From now on, we know how to merge nodes. Now, we need to figure out the most important part of GBUMS, which is the stopping criterion. The stopping criterion is to prevent over-generalisation. In other words, we need to stop the algorithm before getting the initial M-corpus (since no merging is possible). The criterion comes from [14]. This criterion is based on the Bayesian Information Criterion (BIC) and Jensen-Shannon divergence.

BIC is used for selecting a model (set of morphemes) which fits a data set (M-Corpus) without being too complex. BIC is a trade-off between the maximum likelihood, the parameters of the model (probability and length of each morpheme) and the number of elements in the data set (frequency of each morpheme). The smaller the value of BIC is, the better the model is. We use the maximum of the Jensen-Shannon divergence to analyse the increase of log-likelihood between two models. The Jensen-Shannon divergence is defined as follows [5]:

Definition 4. *The* Jensen-Shannon divergence *is defined for two morphemes* m_1 *and* m_2 *as the decrease in entropy between the concatenated and the individual morphemes:*

$$D_{JS}(m_1, m_2) = H(m_1 \cdot m_2) - \frac{L_{m_1} H(m_1) + L_{m_2} H(m_2)}{N} \tag{3}$$

where $H(m) = -P(m) \log_2 P(m)$, $N = \sum_m Freq(m)$ *and* L_m *is the string length of* m.

Stopping criterion *requires that* $\Delta BIC < 0$ *which translates to:*

$$\max_{m_1, m_2} D_{JS}(m_1, m_2) \leq 2 \log_2 N \tag{4}$$

We stress that the BIC is equal to MDL except that the BIC sign is opposite [8]. To sum up, the GBUMS is presented in Algorithm 1.

Algorithm 1. The GBUMS morpheme segmentation algorithm

```
input  M-Corpus = List of Strings
output  M-CorpusSeg = List of Strings
  M-CorpusSeg ← SegmentInLetters(M-Corpus);
  Graph ← InitialiseGraph(M-CorpusSeg);
  repeat
    Max ← 0;
    for all (p,q) ∈ Graph do
      ML_Max ← Morph_Lift(p, q);
      if ML_Max > Max then
        Max ← ML_Max;
        pMax ← p;
        qMax ← q;
      end if
    end for
    Graph ← MergeNodes(Graph, pMax, qMax);
    M-CorpusSeg ← DeleteBoundaries(M-CorpusSeg, Label(pMax), Label(qMax));
    Graph ← AdjustGraph(M-corpusSeg, Graph);
  until StoppingCriterion(pMax, qMax, Max)
```

Note that the M-Corpus is completely segmented at the beginning of the algorithm. Then, the boundaries in the segmented M-Corpus are removed step by step according to a pair found in the graph with the maximum value for $Morph\_Lift$. When the stopping criterion is fulfilled, the segmented M-Corpus represents the morpheme sequences. In section 3, we showed how to extract stems using MDLWS. Here, we presented a method to extract sequences of prefixes and suffixes using GBUMS. The UNGRADE (UNsupervised GRAph DEcomposition) algorithm combines the results found with MDLWS and GBUMS. This aggregation step is straightforward and realized by merging the results of

the prefix analysis with the stem and suffix analysis for each word which were found by MDLWS and GBUMS. The complete algorithm called UNGRADE (UNsupervised GRAph DEcomposition) including all phases is detailed in [11].

5 Processing and Results

In order to test UNGRADE, we used the Morpho Challenge 2009 data sets which include Arabic (non-vowelized 14957 words, vowelized 19244 words), German (1266159 words), Turkish (617298 words), Finnish (2206719 words) and English (384903 words). Before running UNGRADE on the different data sets, we decided to use a pre-processing algorithm to remove marginal words and potential noise. To do so, we analysed the word length distribution to remove infrequent short and long words [11]. The new data sets included Arabic (non-vowelized 14919 words, vowelized 19212 words), German (868219 words), Turkish (534702 words), Finnish (1991275 words) and English (252364 words). Therefore, we used smaller data sets as input to UNGRADE. After running UNGRADE, in order to segment the remaining words of the original data sets, we used a segmented corpus from the output of UNGRADE as a model of segmentation and applied it to all words from the original corpus. Our model was simply a mapping from word to segments (e.g cars becomes c-ar-s). Thus, we did not manage multiple solutions. Also, unseen words without a mapping from our model were left unsegmented. For unseen words, it is clear that there is a search problem which needs a cost function to decide between multiple solutions. This search problem will be investigated in future work.

The evaluation measure used is the *F-measure* which is the harmonic mean of precision and recall [13]. We stress that in our approach morphemes are not labeled by grammatical categories but simply by themselves (i.e., every distinct morpheme has a distinct label). The final results are computed using the Morpho Challenge gold standard and are given in Table 2. The F-measure for German, English, Turkish and Finnish are of the same order of magnitude (between 33.44% and 36.58%). Surprisingly, Arabic non-vowelized provided the worst (26.78%) and Arabic vowelized the best (54.36%) F-measure among all languages. The difference in F-measure for Arabic is explained by the average word length. For Arabic, the non-vowelized words are on average almost half the length for the same number of morphemes.

We note that the precision is higher than recall for all data sets except English. The low level of precision in English is due to the low average number of morphemes. This observation is confirmed in Arabic (vowelized) where the average number of morphemes is higher and therefore gives a high precision. It is interesting to remark that even if the starting point to look for a stem of the UNGRADE algorithm is less correct for Turkish (Turkish does not have prefixes), the results are quite competitive for Finnish. To sum up, UNGRADE is more efficient for a language with long words on average and a high average number of morphemes per word. Analysing the mistakes committed by UNGRADE showed a difficulty to identify words having no prefixes, no suffixes or both. For Turkish,

Table 1. Pre-processing of data sets

Language	Word	Real Segmentation	UNGRADE segmentation		
			Prefix sequence	Stem	Suffix sequence
English	young	young	y	oun	g
	broadcasting	broad-cast-ing	broa	dc	asting
Turkish	canlandIrmIStIr	can-lan-dIr-mIS-tIr	canla-n-dI	rmI	StIr

Table 2. Results from the Morpho Challenge 2009

Language	Precision	Recall	F-Measure
English	.2829	.5174	.3658
German	.3902	.2925	.3344
Finnish	.4078	.3302	.3649
Turkish	.4667	.3016	.3664
Arabic (non-vowelized)	.8348	.1595	.2678
Arabic (vowelized)	.7215	.4361	.5436

we observed that the major problem was the identification of the stem since in most cases it is the first morpheme of each word (Table 1).

6 Conclusion and Future Directions

We have presented UNGRADE, an unsupervised algorithm to decompose words. We assumed that each word for languages contains prefixes, a stem and suffixes without giving a limit on the number of prefixes and suffixes. UNGRADE has been tested on 5 languages (English, German, Finnish, Turkish and Arabic vowelized and Arabic non-vowelized) and results are provided according to a gold standard. UNGRADE gives similar results of F-measure for English, Finnish, German and Turkish with 35.79% of F-measure on average. For Arabic, the results demonstrate that UNGRADE is more efficient with long words including a high number of morphemes (54.36% of F-measure for vowelized against 26.78% for non-vowelized). In future work we will investigate different starting points for the search for stems, e.g. the beginning of the word, the end of the word, etc. Futhermore, a committee could choose the best segmentation under some MDL criterion which may improve the results.

Acknowledgments. We would like to thank our team colleagues Roger Tucker and Ksenia Shalonova for general advice on general morphological analysis. The work was sponsored by EPSRC grant EP/E010857/1 *Learning the morphology of complex synthetic languages*.

References

1. Brent, M.: Minimal generative models: A middle ground between neurons and triggers. In: 15th Annual Conference of the Cognitive Science Society. pp. 28–36 (1993)

2. Brent, M., Murthy, S., Lundberg, A.: Discovering Morphemic Suffixes A Case Stud. In MDL Induction. In: Fifth International Workshop on AI and Statistics. pp. 264–271 (1995)
3. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. In: ACM SIGMOD International Conference on Management of Data, pp. 255–264. ACM, New York (1997)
4. Creutz, M., Lagus, K.: Inducing the morphological lexicon of a natural language from unannotated text. In: AKRR 2005, International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning, Espoo, Finland, pp. 106–113 (June 2005)
5. Dagan, I., Lee, L., Pereira, F.: Similarity-Based Methods for Word Sense Disambiguation. In: Thirty-Fifth Annual Meeting of the ACL and Eighth Conference of the EACL, pp. 56–63 (1997)
6. Déjean, H.: Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora. In: Joint Conferences on NeMLaP3 and CoNLL, pp. 295–298 (1998)
7. Gaussier, E.: Unsupervised Learning of Derivational Morphology From Inflectional Lexicons. In: ACL 1999 Workshop on Unsupervised Learning in NLP. ACL (1999)
8. Geng, Y., Wu, W.: A Bayesian Information Criterion Based Approach for Model Complexity Selection in Speaker Identification. In: International Conference on Advanced Language Processing and Web Information Technology, pp. 264–268. IEEE Computer Society Press, Los Alamitos (July 2008)
9. Goldsmith, J.: Unsupervised Learning of the Morphology of a Natural Language. Computational Linguistics 27(2), 153–198 (2001)
10. Golénia, B.: Learning rules in morphology of complex synthetic languages. Master's thesis, University of Paris V (2008)
11. Golénia, B., Spiegler, S., Flach, P.: UNGRADE: UNsupervised GRAph DEcomposition. In: Working Notes for the CLEF 2009 Workshop, CLEF 2009, Corfu, Greece (2009)
12. Harris, Z.: From Phoneme to Morpheme. Language 31(2), 190–222 (1955), http://dx.doi.org/10.2307/411036
13. Kurimo, M., Virpioja, S., Turunen, V., Blackwood, G., Byrne, W.: Overview and Results of Morpho Challenge 2009. In: Multilingual Information Access Evaluation 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, Revised Selected Papers, September 30 - October 2, 2009. LNCS, vol. I, Springer, Heidelberg (2010)
14. Li, W.: New Stopping Criteria for Segmenting DNA Sequences. Physical Review Letters 86(25), 5815–5818 (2001)
15. Monson, C., Carbonell, J., Lavie, A., Levin, L.: ParaMor: Finding Paradigms across Morphology. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 900–907. Springer, Heidelberg (2008)
16. Monson, C., Carbonell, J., Lavie, A., Levin, L.: Paramor: Finding Paradigms across Morphology. In: Advances in Multilingual and Multimodal Information Retrieval, pp. 900–907. Springer, Heidelberg (2008)
17. Shalonova, K., Golénia, B., Flach, P.: Towards Learning Morphology for Under-Resourced Fusional and Agglutinating Languages. IEEE Transactions on Audio, Speech, and Language Processing 17(5), 956–965 (2009)
18. Spiegler, S., Golénia, B., Shalonova, K., Flach, P., Tucker, R.: Learning the morphology of Zulu with different degrees of supervision. In: Spoken Language Technology Workshop, 2008. SLT 2008, pp. 9–12. IEEE, Los Alamitos (December 2008)

Clustering Morphological Paradigms Using Syntactic Categories

Burcu Can and Suresh Manandhar

Department of Computer Science, University of York,
York, YO10 5DD, United Kingdom
{burcu,suresh}@cs.york.ac.uk

Abstract. We propose a new clustering algorithm for the induction of the morphological paradigms. Our method is unsupervised and exploits the syntactic categories of the words acquired by an unsupervised syntactic category induction algorithm [1]. Previous research [2,3] on joint learning of morphology and syntax has shown that both types of knowledge affect each other making it possible to use one type of knowledge to help learn the other one.

1 Introduction

Morphological analysis has been an important subtask for most Natural Language Processing (NLP) problems such as Machine Translation, Question Answering, Information Retrieval, Part-of-Speech Tagging etc. For languages having an agglutinative morphology such as Finnish, and Turkish, it would require a massive effort to create a dictionary with the all possible word forms in the language. For example, in Turkish, the verb *oku-mak* which means *to read*, can take: *oku-yor* (*he/she is reading*), *oku-yor-um* (*I am reading*), *oku-r* (*he/she reads* or *the reader*), *oku-maz* (*he/she does not read*), *oku-n-mak* (*to be read*), *oku-n-du* (*it was read*) etc. These are, however, only a small number of the word formations of the word *oku*. For example, in Finnish, if all the derivational and inflectional morphemes are considered, the number of the different forms of a verb can be in thousands [4].

Unsupervised morphology learning is a challenging problem especially for agglutinative languages. Initial foundational work on unsupervised morphology learning is due to Harris [5]. Harris' work was based on exploiting the distributional signature of the character-character bigrams within words. Since then, there has been a huge amount of progress made in the field. These can be classified into the following approaches: Minimum Description Length (MDL) model [6,7,8], Letter Successor Variety (LSV) model [9], Semantic models (ex: Latent Semantic Analysis - LSA) [10], Probabilistic models [11,12], and Paradigmatic models [13].

The strong correspondence between morphological and PoS tag information, makes it possible to use one type of knowledge to help learn the other. For example, most adverbs end with *-ly*, most verbs in past tense form ends with *-ed*, and so on. Therefore it is more effective to treat this process as a joint learning

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 641–648, 2010.

problem. Hu et. al. [3] extend Goldsmith's model [8] by using the Part-of-Speech tags assigned by the Tree Tagger [14] and explore the link between morphological signatures and PoS tags. However, their method is not fully unsupervised due to the use of a supervised PoS tagger. Another method [2] uses fixed endings of the words for PoS clustering. Although the characters sequences used are not morphologically meaningful, their results show that these sequences also help in PoS clustering.

In this paper, we describe an unsupervised morphological paradigm induction method that exploits the syntactic categories induced by a syntactic category clustering algorithm. The system was evaluated in Morpho Challenge 2009 where proposed morphological analyses were compared to a gold standard. Section 2 describes the model in detail, Section 3 gives the evaluation results of Morpho Challenge 2009, and finally Section 4 presents a discussion of the model and addresses the future work.

2 Morphological Paradigms through Syntactic Clusters

In this section, we describe our method step by step.

2.1 Syntactic Category Induction

We use context distribution clustering algorithm due to Clark [1]. In this approach, a context for a word is defined as the pair *<left context word, right context word>*. Thus, in the sentence, *"John likes Mary"*, the context pair for *"likes"* will be the pair *<John, Mary>*. Using the corpus supplied from the Morpho Challenge 2009 and the Cross Language Evaluation Forum (CLEF) 2009, we generate the context vectors for every word. These vectors are clustered using average link clustering and Kullback-Leibler (KL) divergence as the distance metric. It also should be emphasised that any other unsupervised syntactic category induction method can be replaced with this algorithm without affecting the method presented in this paper.

We cluster words into 77 syntactic categories which is the number of the tags defined in CLAWS tagset used for tagging the British National Corpus (BNC). Initial clusters are created by taking 77 most frequent words in the corpus. The same number of clusters are used for Turkish and German. The resulting clusters are the reflections of the mirror syntactic categories such as verbs in past tense form, verbs in progressive form, nouns, comparative adjectives, adverbs, and so on.

2.2 Identifying Potential Morphemes

Each syntactic cluster includes a set of potential morphemes produced by splitting each word into all possible stem-suffix combinations. For each potential morpheme we calculate its conditional probability $p(m|c)$ where m denotes the morpheme, and c denotes the cluster. When the potential morphemes are ranked

according to their conditional probabilities, only those above a threshold value (see Evaluation and Results) are considered in the following step. This ranking is used to eliminate the potential non-morphemes with a low conditional probability.

2.3 Capturing Morphological Paradigms

Our definition of a paradigm deviates from that of Goldsmith [8] due to the addition of syntactic categories. In our framework, each morpheme is tied to a syntactic cluster. More precisely, a paradigm P is a list of morpheme/cluster pairs together with a list of stems: $P = < \{m_1/c_1, \ldots, m_r/c_r\}, \{s_1, \ldots, s_k\} >$ where m_i denotes a morpheme belonging to the cluster c_i, and s_j denotes a stem such that $\forall m_i/c_i \in \{m_1/c_1, \ldots, m_r/c_r\}$ and $\forall s_j \in \{s_1, \ldots, s_k\} : s_j + m_i \in c_i$.

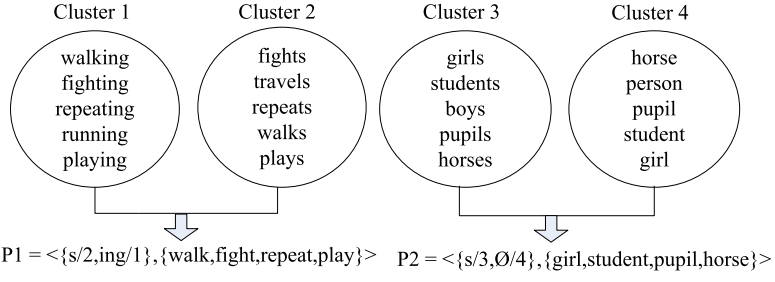


Fig. 1. A sample set of syntactic clusters, and the potential morphemes in each cluster

In each iteration, a potential morpheme pair across two different syntactic clusters with the highest number of common stems is chosen for merging. Once a morpheme pair is merged, the words that belong to this newly formed paradigm are removed from their respective clusters (see Fig. 1). This forms the basis of the paradigm-capturing mechanism. We assume that a word can only belong to a single morphological paradigm. The above procedure is repeated until no further paradigms are created.

2.4 Merging Paradigms

In this step, the initial paradigms are merged to create more general paradigms. The decision to merge two paradigms P_1, P_2 is based on the expected accuracy of the merged paradigm. The expected paradigm accuracy is defined as the ratio of the common stems to the total number of stems included in the two paradigms. More precisely, given paradigms P_1, P_2, let S be the total number of common stems. Let N_1 be the total number of stems in P_1 that are not present in P_2 and N_2 vice versa. Then, we compute the expected paradigm accuracy by:

Algorithm 1. Algorithm for paradigm-capturing using syntactic categories

```
Apply syntactic category induction algorithm to the input corpus.
Generate all possible morphemes by splitting the words in all possible stem-suffix combinations.
For each cluster c and morpheme m, compute maximum likelihood estimates of p(m | c).
Keep all m (in c) with p(m | c) > t, where t is a threshold.
repeat
    for all Clusters c1, c2 do
        Pick morphemes m1 in c1 and m2 in c2 with the highest number of common stems.
        Store P = {m1/c1, m2/c2} as the new paradigm.
        Remove all words in c1 with morpheme m1 and associate these words with P.
        Remove all words in c2 with morpheme m2 and associate these words with P.
    end for
    for each paradigm pair P1, P2 such that Acc(P1, P2) > T, where T is a threshold do
        Create new merged paradigm P = P1 ∪ P2.
        Associate all words from P1 and P2 with P.
        Delete paradigms P1, P2. .
    end for
until No morpheme pair consisting of at least one common stem is left
```

$$Acc_1 = \frac{S}{S+N_1},\ Acc_2 = \frac{S}{S+N_2},\ Acc = \frac{Acc_1+Acc_2}{2} \tag{1}$$

Algorithm 1 describes the complete paradigm-capturing process.

During each iteration, the paradigm pair having an expected accuracy greater than a given threshold value (see Evaluation and Results) is merged. Once two paradigms are merged, stems that occur in only one of the paradigms inherit the morphemes from the other paradigm. This mechanism helps create more general paradigms and recover missing word forms. As we see from the example (given in Figure 2), although the words *complements*, *complement*, *betrayed*, *betraying*, *altered*, *altering*, *finding* do not exist in the corpus, with the proposed paradigm-merging mechanism non-occurring forms of the words are also captured.

2.5 Morphological Segmentation

To segment a given word using the learnt paradigms we follow the following procedure. Words already included in the paradigms are simply segmented by using the morpheme set in the paradigm. For words that are not included in the paradigms, a morpheme dictionary is created with the morphemes in all paradigms. Therefore, unknown words are segmented with the longest morpheme available in the morpheme dictionary recursively. For compound words (e.g. the word *hausaufgaben* including words *haus*, *auf* and *gaben* in German), the same

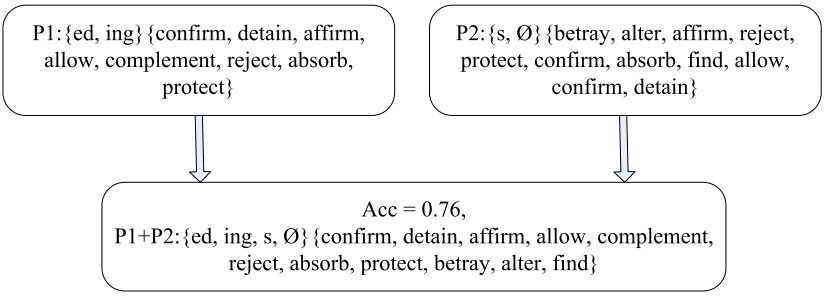


Fig. 2. A sample of paradigm-merging procedure

Algorithm 2. Morphological Segmentation

1: **for all** For each given word, w, to be segmented **do**
2: **if** w already exists in a paradigm P **then**
3: Split w using P as $w = u + m$
4: **else**
5: $u = w$
6: **end if**
7: If possible split u recursively from the rightmost end by using the morpheme dictionary as $u = s_1 + \ldots + s_n$ otherwise $s_1 = u$
8: If possible split s_1 into its sub-words recursively from the rightmost end as $s_1 = w_1 + \ldots + w_n$
9: **end for**

recursive approach is applied by using the corpus as a word dictionary. The algorithm for the segmentation of the words is given in Algorithm 2.

3 Evaluation and Results

The model has been evaluated in the Morpho Challenge 2009 competition. The corpus from Morpho Challenge 2009 and the Cross Language Evaluation Forum (CLEF) 2009 were used for training our system on 3 different languages: English, German and Turkish. For the initial clustering, corpora provided in Morpho Challenge 2009[1] were used. For clustering the words in the word list to be segmented, for English and German, datasets supplied by the CLEF organization[2] were used. For Turkish, we made use of manually collected newspaper archives.

[1] http://www.cis.hut.fi/morphochallenge2009

[2] http://www.clef-campaign.org/. English datasets: Los Angeles Times 1994 (425 mb), Glasgow Herald 1995 (154 mb). German datasets: Frankfurter Rundschau 1994 (320 mb), Der Spiegel 1994/95 (63 mb), SDA German 1994 (144 mb), SDA German 1995 (141 mb).

Table 1. Evaluation results of the Morpho Challenge Competition 1

Language	Precision(%)	Recall(%)	F-measure(%)	F/Winner(%)
English	58.52	44.82	50.76	62.31
German	57.67	42.67	49.05	56.14
Turkish	41.39	38.13	39.70	53.53

Table 2. Obtained average precisions (AP) for the Morpho Challenge Competition 2

Language	AP(%)	AP(%)-Winner
English	0.2940	0.3890
German	0.4006	0.4490

Although our model is unsupervised, two prior parameters are required to be set: t for the conditional probability $p(m|c)$ of the potential morphemes and T for the paradigm accuracy threshold for merging the paradigms. We set t=0.1 and T=0.75 in all the experiments.

The system was evaluated in Competition 1 & 2 of Morpho Challenge 2009. In Competition 1, proposed analyses are compared to a gold standard analysis of a word list. Details of the tasks and evaluation can be found in the Overview and Results of Morpho Challenge 2009 [15]. Evaluation results corresponding to the Competition 1 are given in Table 1.

In the Competition 2, proposed morphological analyses are used in an information retrieval task. To this end, words are replaced with their morphemes. Our results for German had an average precision of 0.4006% whereas the winning system [16] had an average precision of 0.4490%. Our results for English had an average precision of 0.2940% whereas the winning system [17] had an average precision of 0.3890% (see Table 2).

4 Discussion and Conclusions

To our knowledge, there has been limited work on the combined learning of syntax and morphology. In Morpho Challenge 2009, our model is the only system making use of the syntactic categories. Morphology is highly correlated with the syntactic categories of words. Therefore, our system is able to find the potential morphemes by only considering conditional probabilities $p(m|c)$. A list of highest-ranked potential morphemes belonging to different syntactic categories are given in Table 3 for English, German and Turkish.

The paradigm including the most number of stems for English has the morpheme set {*s, ing, ed,* $\emptyset$} where $\emptyset$ denotes the NULL suffix, for Turkish it has {*u, a, e, i*}, and for German it has {*er,* $\emptyset$*, e, en*}. Our paradigm merging method is able to compensate for the missing forms of the words. For example, as shown in Fig. 2, although the words such as *altering*, and *finding* do not exist in the real corpus, they are produced during the merging. However, our system still requires a large dataset for syntactic category induction. We only consider words

Table 3. Examples for high ranked potential morphemes in clusters

English		German		Turkish	
Cluster	Morphemes	Cluster	Morphemes	Cluster	Morphemes
1	-s	1	-n,-en	1	-i,-si,-ri
2	-d,-ed	2	-e,-te	2	-mak,-mek,-mesi,-masi
3	-ng,-ing	3	-g,-ng,-ung	3	-an,-en
4	-y,-ly	4	-r,-er	4	-r,ar,er,-ler,-lar
5	-s,-rs,-ers	5	-n,-en,-rn,-ern	5	-r,-ir,-dir,-ır,-dır
6	-ing,-ng,g	6	-ch,-ich,-lich	6	-e,-a

having a frequency greater than 10 to eliminate noise. To segment non-frequent words we propose a heuristic method based on using a morphmeme dictionary. However, the usage of such a morpheme dictionary can often have undesirable results. For example, the word *beer* is forced to be segmented as *be-er* due to the morpheme *er* found in the dictionary.

Our model allows more than one morpheme boundary. This makes our system usable for the morphological analysis of the agglutinative languages. For example, in Turkish, the word *çukurlarıyla* (which means *"with their burrows"*) has the morpheme boundaries: *çukur-lar-ı-y-la*. However, in our heuristic method, the use of morpheme dictionary causes undesirable results. For example, the same word *çukurlarıyla* is segmented by our method as: *çu-kurları-y-la*.

Finally, our system is sensitive to the thresholds we set for 1. identifying potential morphemes and 2. expected paradigm accuracy. In future work, we hope to address these and previously mentioned deficiencies.

Despite the observed deficiencies, we obtained promising results in Morpho Challenge 2009. Our precision and recall values are balanced and undersegmentation is not very prominent for all languages that we evaluated. We believe that our work clearly demonstrates that joint modeling of syntactic categories and morphology is the key for building successful morphological analysis system. In addition, our work demonstrates how morphological paradigms can be learnt by taking advantage of the syntactic information.

References

1. Clark, A.S.: Inducing syntactic categories by context distribution clustering. In: Proceedings of CoNLLL 2000 and LLL 2000, Morristown, NJ, USA, pp. 91–94. ACL (2000)
2. Clark, A.S.: Combining distributional and morphological information for part of speech induction. In: EACL 2003: Proceedings of the 10th EACL, Morristown, NJ, USA, pp. 59–66. ACL (2003)
3. Hu, Y., Matveeva, I., Goldsmith, J., Sprague, C.: Using morphology and syntax together in unsupervised learning. In: Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition, Ann Arbor, Michigan, June 2005, pp. 20–27. ACL (2005)

4. Karlsson, F.: Finnish grammar. WSOY, Juva (1983)
5. Harris, Z.S.: Distributional structure. Word 10(23), 146–162 (1954)
6. Brent, M.R., Murthy, S.K., Lundberg, A.: Discovering morphemic suffixes a case study in mdl induction. In: Fifth International Workshop on AI and Statistics, pp. 264–271 (1995)
7. Creutz, M., Lagus, K.: Unsupervised discovery of morphemes. In: Proceedings of the ACL-02 workshop on Morphological and phonological learning, Morristown, NJ, USA, pp. 21–30. ACL (2002)
8. Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Computational Linguistics 27(2), 153–198 (2001)
9. Bordag, S.: Unsupervised and knowledge-free morpheme segmentation and analysis. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 881–891. Springer, Heidelberg (2008)
10. Schone, P., Jurafsky, D.: Knowledge-free induction of morphology using latent semantic analysis. In: Proceedings of CoNLL 2000 and LLL 2000, Morristown, NJ, USA, pp. 67–72. ACL (2000)
11. Snover, M.G., Jarosz, G.E., Brent, M.R.: Unsupervised learning of morphology using a novel directed search algorithm: Taking the first step. In: Proceedings of the ACL 2002 Workshop on Morphological and Phonological Learning, Morristown, NJ, USA, pp. 11–20. ACL (2002)
12. Creutz, M.: Unsupervised segmentation of words using prior distributions of morph length and frequency. In: ACL 2003: Proceedings of the 41st ACL, pp. 280–287, Morristown, NJ, USA. ACL (2003)
13. Monson, C.: Paramor: From Paradigm Structure to Natural Language Morphology Induction. PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University (2008)
14. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK (1994)
15. Kurimo, M., Virpioja, S., Turunen, V.T., Blackwood, G.W., Byrne, W.: Overview and results of morpho challenge 2009. In: Multilingual Information Access Evaluation Vol. I 10th Workshop of the CLEF 2009, Corfu, Greece, September 30 - October 2, Revised Selected Papers. Springer, Heidelberg (2009)
16. Monson, C., Hollingshead, K., Roark, B.: Probabilistic paramor. In: Working Notes for the CLEF Workshop, Corfu, Greece (2009)
17. Lignos, C., Chan, E., Marcus, M.P., Yang, C.: A rule-based unsupervised morphology learning framework. In: Working Notes for the CLEF Workshop, Corfu, Greece (2009)

Simulating Morphological Analyzers with Stochastic Taggers for Confidence Estimation

Christian Monson, Kristy Hollingshead, and Brian Roark

Center for Spoken Language Understanding, Oregon Health & Science University
monsonc@csee.ogi.edu, hollingk@cslu.ogi.edu, roark@cslu.ogi.edu

Abstract. We propose a method for providing stochastic confidence estimates for rule-based and black-box natural language (NL) processing systems. Our method does not require labeled training data: We simply train stochastic models on the output of the original NL systems. Numeric confidence estimates enable both minimum Bayes risk–style optimization as well as principled system combination for these knowledge-based and black-box systems. In our specific experiments, we enrich ParaMor, a rule-based system for unsupervised morphology induction, with probabilistic segmentation confidences by training a statistical natural language tagger to simulate ParaMor's morphological segmentations. By adjusting the numeric threshold above which the simulator proposes morpheme boundaries, we improve F_1 of morpheme identification on a Hungarian corpus by 5.9% absolute. With numeric confidences in hand, we also combine ParaMor's segmentation decisions with those of a second (black-box) unsupervised morphology induction system, Morfessor. Our joint ParaMor-Morfessor system enhances F_1 performance by a further 3.4% absolute, ultimately moving F_1 from 41.4% to 50.7%.

1 Background

The Importance of Confidence Estimation. Confidence estimation, one of the key benefits of probabilistic modeling of natural language (NL), is crucial both for minimum Bayes risk inference as well as for stochastic system combination. Minimum Bayes risk inference enables the tuning of NL systems to achieve high precision, high recall, or something in between, while system combination can unite the complementary strengths of independent systems. Unfortunately, NL systems that we would like to optimize or combine do not always produce weights from which confidence estimates may be calculated. In some domains, knowledge-based systems are widely used and are effective, e.g., the best stemming, tokenization, and morphological analyzers for many languages are hard clustering approaches that do not involve weights or even yield alternative analyses. For other tasks, weights may be used system-internally, but are not immediately accessible to the end-user—such a system is a black-box to the user.

Here, we investigate simulating knowledge-based and black-box systems with stochastic models by training in a supervised manner on the output of the non-stochastic (or black-box) systems. The specific systems that we mimic are unsupervised morphological analyzers, which we simulate via discriminatively trained character taggers.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 649–657, 2010.

The taggers' easily accessible posterior probabilities can serve as confidence measures for the original systems. Leveraging these newfound confidence scores, we pursue minimum Bayes risk–style thresholding of tags (for higher morpheme recall) as well as principled system combination approaches (for higher overall accuracy). As an added benefit, the shallow tag-and-character features that are employed by our simulation taggers enable generalization from the baseline system—the simulation taggers make correct decisions in contexts where the original systems do not.

A Brief History of Unsupervised Morphology Induction. Unsupervised morphology induction is the task of learning the morphological analyses of the words of an arbitrary natural language from nothing more than a raw corpus of unannotated text. Analyzing words down to the morpheme level has helped natural language processing tasks from machine translation [1] to speech recognition [2]. But building a morphological analysis system by hand can take person-months of time—hence the need for automatic methods for morphology induction.

Many approaches to unsupervised morphology induction have been proposed. Techniques inspired by Zellig Harris [3] measure the probabilities of word-internal character transitions to identify likely morpheme boundaries [4]. Other systems rely on the minimum description length principle to pick out a set of highly descriptive morphemes [5], [2]. Recent work on unsupervised morphology induction for Semitic languages has focused on estimating robust statistical models of morphology [6], [7]. And this paper extends ParaMor,[1] an induction system that leverages morphological paradigms as the inherent structure of natural language morphology [8].

Section 2 introduces the baseline systems that our taggers simulated together with the specific tagging approach that we take; Section 3 presents empirical results, demonstrating the ultimate utility of simulation; while Section 4 concludes.

2 Simulating Morphological Analyzers

The ParaMor system for unsupervised morphology induction builds sets of suffixes that model the paradigm structure found in natural language inflectional morphology. ParaMor competed in both the 2007 and 2008 Morpho Challenge Competitions [9], both solo and in a joint submission with a second unsupervised morphology induction system Morfessor [2]. Setting aside the joint ParaMor-Morfessor submission, the solo ParaMor placed first in the 2008 Turkish Linguistic competition, 46.5% F_1, and second in English, at 52.5% F_1. Meanwhile the joint ParaMor-Morfessor system placed first overall in the 2008 Linguistic competitions for German, Finnish, Turkish, and Arabic. ParaMor's successes are particularly remarkable given that ParaMor is a rule-based system incapable of measuring the confidence of the morphological segmentations it proposes—without a confidence measure, ParaMor cannot optimize its segmentation strategy toward any particular metric.

Simulating ParaMor with a Statistical Model. To gain the advantages that stochastic confidence measures provide, while retaining the strengths of the ParaMor morphology induction algorithm, we train a statistical model to simulate ParaMor's morphological

[1] ParaMor is freely available from: `http://www.cslu.ogi.edu/~monsonc/ParaMor.html`

segmentations. Specifically, we view the morphology segmentation task as a labeling problem akin to part-of-speech tagging. Statistical tagging is a proven and well-understood natural language processing technique that has been adapted to a variety of problems beyond part-of-speech labeling. Taggers have been used for named entity recognition [10] and NP-chunking [11]; to flag words on the periphery of a parse constituent [12]; as well as to segment written Chinese into words [13]—a task closely related to morphology segmentation.

We trained a finite-stage tagger [14] to identify, for each character, *c*, in a given word, whether or not ParaMor would place a morpheme boundary immediately before *c*. We supplied the tagger with three types of features: 1. One-sided character n-grams, 2. Two-sided character n-grams, and 3. Morpheme-tag n-grams. The one-sided n-grams are the uni-, bi-, tri-, and 4-grams that either end or begin with *c*; The two-sided n-grams are all 7-grams that extend up to five characters to the left or right of *c*; And the morpheme-tag features are the unigram, bigram, and trigram morpheme-tags, covering the current and two previous tags.

We used the averaged perceptron algorithm [15] to train the tagger. During training, the decoding process is performed using a Viterbi search with a second-order Markov assumption. At test-time, we use the forward-backward algorithm, again with a second-order Markov assumption, to output the perceptron-score of each morphological tag for each character in the word. The main benefit of decoding in this manner is that, by normalizing the scores at each character (using softmax due to the log linear modeling), we can extract the posterior probability of each tag at each character rather than just the single perceptron-preferred solution for the entire word.

Fidelity. Using our finite state tagger, we construct a baseline ParaMor-simulated segmentation by placing morpheme boundaries before each character that is tagged as the start of a new stem or affix with a posterior probability greater than 0.5. This baseline mimic segmentation, although trained to emulate ParaMor's segmentations, will not be identical to ParaMor's original segmentations of a set of words. Table 1 summarizes our tagging accuracy at emulating segmentations for the five languages and six data sets of the Linguistic competition of Morpho Challenge 2009 [16]. Tagging accuracy, the percentage of correctly tagged characters, is the standard metric used to evaluate performance in the tagging literature. We calculate accuracy by averaging over held-out test folds during 10-fold cross validation. Resource constraints compelled us to divide the full Morpho Challenge data for each language into disjoint subsets each containing approximately 100,000 word types. We then trained separate taggers over each data subset, and accuracy numbers are averaged over all subsets.

For all the test languages and scenarios, our tagger successfully emulates ParaMor at a tagging accuracy above 93%, with particular strength on German, 96.6%, and English, 97.6%. Tagging accuracies in the mid 90%s are comparable to accuracies reported for other tagging tasks.

Table 1. Tagging accuracy at simulating ParaMor's morphological segmentations

English	German	Finnish	Turkish	Arabic -V	Arabic +V
97.6%	96.6%	93.5%	93.6%	93.3%	93.7%

Generalization. The mimic tagger's departures from the original ParaMor segmentation may either hurt or improve the segmentation quality. On the one hand, when the mimic tagger deviates from the ParaMor segmentation, the mimic may be capturing some real generalization of morphological structure that is hidden in the statistical distribution of ParaMor's original segmentation. On the other hand, a disagreement between the original and the simulated ParaMor segmentations may simply be a failure of the tagger to model the irregularities inherent in natural language morphology.

To evaluate the generalization performance of our ParaMor tagging simulator, we performed a development evaluation over a Hungarian dataset. We used Hunmorph [17], a hand-built Hungarian morphological analyzer, to produce a morphological answer key containing 500,000 unique Hungarian word types from the Hunglish corpus [18]. Our Hungarian ParaMor tagger mimic successfully generalizes: Where the original ParaMor attained an F_1 of 41.4%, the ParaMor simulator improved F_1 to 42.7%, by virtue of slightly higher recall; this improvement is statistically significant at the 95% confidence level, assuming normally distributed F_1 scores.

Optimization. Having retained ParaMor's underlying performance quality by training a natural language tagger to simulate ParaMor's segmentations, we next increase F_1 further by leveraging the tagger's probabilistic segmentation scores in a minimum Bayes risk–style optimization procedure, as follows:

1. For each word, w, in a corpus
   For each character, c, that does not begin w
   Record, in a list, L, the tagger mimic's probability that c begins a morpheme.
2. Sort the probabilities in L
3. Assign k to be the number of probability scores that are larger than 0.5
4. For a given positive factor, α, identify in L the probability score, S, above which αk of the probabilities lie
5. Segment at characters which receive a probabilistic segmentation score above S

In prose, to trade off recall against precision, we move the probability threshold from the default of 0.5 to that value which will permit αk segmentations. Given the extremely peaked probability scores that the mimic tagger outputs, we adjust the number of segmentation points via α rather than via the probability threshold directly.

The Linguistic competition of Morpho Challenge evaluates morphological segmentation systems using precision, recall, and F_1 of morpheme identification. Fig. 1 plots the precision, recall, and F_1 of the ParaMor tagger mimic as the number of word-internal morpheme boundaries varies between one half and four times the baseline k number of word-internal boundaries. As Fig. 1 shows, adjusting α allows for a smooth tradeoff between precision and recall. F_1 reaches its maximum value of 47.5% at $\alpha = 4/3$. As is typical when trading off precision against recall, the maximum F_1 occurs near the α location where recall overtakes precision. The improvement in F_1 for the ParaMor tagger mimic of 4.8% is statistically significant at a 95% confidence.

System Combination of with Morfessor. In addition to enabling optimization of the ParaMor tagging simulator, numeric confidence scores permit us to combine segmentations derived from ParaMor with segmentations obtained from the freely available unsupervised morphology induction system Morfessor Categories-MAP [2]. In brief, Morfessor searches for a segmentation of a corpus that maximizes the corpus

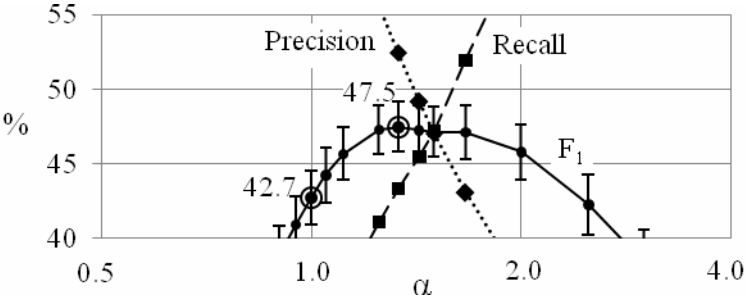


Fig. 1. Precision, Recall, and F_1 of the Hungarian ParaMor tagger mimic as α moves between 0.5 and 4.0. The error bars are 95% confidence intervals on each F_1 value.

probability score according to a specific generative probability model. The Morfessor system then further refines the morphological segmentations it proposes by restricting morpheme sequences with a Hidden Markov Model which permits only (prefix* stem suffix*)+ sequences. Our combined ParaMor-Morfessor systems differ substantially from the ParaMor-Morfessor systems that the lead author submitted to Morpho Challenge 2008—both of our updated combinations merge the ParaMor and the Morfessor segmentations of each word into a *single* analysis.

Joint ParaMor-Morfessor Mimic. The first of our combined ParaMor-Morfessor submissions builds on the idea of tagger mimics. While Morfessor has itself a statistical model that internally scores individual morphological segmentations, the final segmentations that Morfessor proposes are not by default annotated with confidences. Hence, we followed the procedure outlined in Section 2 to train a natural language tagger to simulate Morfessor's morphological analyses. It is encouraging that our technique for inducing probabilities through a mimic tagger immediately extends from a non-statistical system like ParaMor to the black-box scenario for Morfessor.

With separate taggers now simulating both ParaMor and Morfessor we then sum, for each character, *c*, in each word, the tag probabilities from the ParaMor mimic with the corresponding probabilities from the Morfessor mimic. We weighted the probability scores from the ParaMor mimic and the Morfessor mimic equally. To obtain the final morphological segmentation of each word, our combined ParaMor-Morfessor mimic followed the methodology described in Section 2 of optimizing F_1 against our Hungarian development set, with one caveat. Because we weighted the probabilities of ParaMor and Morfessor equally, any segmentation point that is strongly suggested by only one of the two systems receives an adjusted probability score just less than 0.5. Hence, we moved the baseline probability threshold from 0.5 to 0.49. With this single adjustment, the α factor that maximized Hungarian F_1 was 10/9, an 11% increase in the number of proposed morpheme boundaries.

ParaMor-Morfessor Union. The second of the two system combinations that we submitted to Morpho Challenge 2009 fuses a single morphological segmentation from the disparate segmentations proposed by the baseline ParaMor and Morfessor systems by segmenting each word at *every* location that either ParaMor or Morfessor suggests. Hence, this submission is the union of all segmentation points that are proposed by

ParaMor and Morfessor. As an example union segmentation, take the English word *polymers'*: ParaMor's segmentation of this word is *polym +er +s'*, Morfessor's is *polymer +s +'*, and the union analysis: *polym +er +s +'*.

3 Results

ParaMor in Morpho Challenge 2009. To evaluate our ParaMor tagging simulator, we competed in all the language tracks of all three competitions of Morpho Challenge 2009. Here we focus on the results of the Linguistic competition, see [16] for full details on ParaMor's performance at Morpho Challenge 2009.

To analyze morphology in a purely unsupervised fashion, a system must freeze all free parameters across all language tracks of Morpho Challenge. Our tagging mimic systems have one free parameter, α. For all languages of the Linguistic, Information Retrieval, and Machine Translation competitions of Morpho Challenge we set α at that setting which produced the highest F_1 Linguistic score on our Hungarian development set: 4/3 in the case of the ParaMor stand-alone mimic; and 10/9 for the joint ParaMor-Morfessor mimic.

The top two rows of Table 2 contain the precision, recall, and F_1 scores for the original ParaMor, which competed in the 2008 Challenge, and for the ParaMor Tagger Mimic on the non-Arabic languages[2] of Morpho Challenge 2009. Across the board, the gap between precision and recall is smaller for the ParaMor Mimic than it is for the 2008 ParaMor system. In all languages but English, the reduced precision-recall gap results in a higher F_1 score. The increase in F_1 for German, Finnish, and Turkish is more modest than the Hungarian results had led us to hope—about one percentage point in each case. Three reasons likely limited the improvements in F_1. First, the performance rose by a smaller amount for the Challenge test languages than they did for our Hungarian development set because we were explicitly tuning our α parameter to Hungarian. Second, it may be atypical that the tagger mimic generalized

Table 2. (P)recision, (R)ecall, and F_1 scores for the ParaMor and Joint ParaMor-Morfessor systems. *For English, Joint ParaMor-Morfessor achieved its highest F_1 in 2007.

	English			German			Finnish			Turkish		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
ParaMor 2008	63.3	52.0	**57.1**	57.0	42.1	48.4	50.0	37.6	42.9	57.4	45.8	50.9
ParaMor Mimic	53.1	59.0	55.9	50.8	47.7	**49.2**	47.2	40.5	**43.6**	49.5	54.8	**52.0**
Union	55.7	62.3	**58.8**	52.3	60.3	**56.1**	47.9	51.0	**49.4**	47.3	60.0	52.9
Joint Mimic	54.8	60.2	57.4	51.1	57.8	54.2	51.8	45.4	48.4	48.1	60.4	**53.5**
Joint from 2008	70.1*	67.4*	68.7*	64.1	61.5	62.8	65.2	50.4	56.9	66.8	58.0	62.1

[2] The small size of the Arabic training data as well as Arabic's use of morphological processes other than suffixation caused the underlying ParaMor algorithm to suffer extraordinarily low recall in both the vowelized and unvowelized Arabic scenarios.

to outperform the baseline ParaMor system on the Hungarian data. Third, time and resources constrained us to train the tagging simulators over subsets of the full Morpho Challenge data, anecdotally lowering tag-mimic accuracy by a percentage point.

The Joint ParaMor-Morfessor Systems. The bottom three rows of Table 2 summarize the performance of three combined ParaMor-Morfessor systems. The third and fourth rows of Table 2 give performance numbers for, respectively, the ParaMor-Morfessor Union system and for the Joint ParaMor-Morfessor Tagging Mimic. The final row of Table 2 lists the performance of the Joint ParaMor-Morfessor system that was submitted by the lead author to Morpho Challenge 2008.

Although the Union and Joint Mimic systems do outperform at F_1 the solo ParaMor Mimic, it was disappointing that the simple Union outscored the ParaMor-Morfessor Tagger Mimic in three of the four relevant language scenarios. Particularly surprising is that the recall of the Joint Mimic falls below the recall of the Union system in every language but Turkish. With an α factor above 1, the Joint Tagger Mimic is proposing all the segmentation points that either the ParaMor Mimic or the Morfessor Mimic hypothesize—effectively the union of the mimic systems. And yet recall is below that of the raw Union system. We can only conclude that the cumulative failure of the ParaMor and Morfessor Mimics to emulate, let alone generalize from, the original systems' segmentations drags down the recall (and precision) of the Joint Mimic.

Table 2 also highlights the relative success of the 2008 Joint ParaMor-Morfessor system. In particular, the precision scores of the 2008 system are significantly above the precision scores of the Joint Mimic and Union systems that we submitted to the 2009 Challenge. The 2008 system did *not* form a single unified segmentation for each word, but instead simply proposed the ParaMor analysis of each word alongside the Morfessor analysis—as if each word were ambiguous between a ParaMor and a Morfessor analysis. The evaluation procedure of Morpho Challenge performs a non-trivial average over alternative segmentations of a word. It is a shortcoming of the Morpho Challenge evaluation methodology to inflate precision scores when disparate systems' outputs are proposed as 'alternative' analyses.

4 Summary and Next Steps

Using a statistical tagging model we have imbued rule-based and black-box morphology analysis systems with confidence scores. These probabilistic scores have allowed us to successfully optimize the systems' morphological analyses toward a particular metric of interest, the Linguistic evaluation metric of Morpho Challenge 2009.

Looking forward, we believe our statistical taggers can be enhanced along two separate avenues. First, via careful feature engineering: Tagging accuracy might improve by, for example, employing character n-grams longer than 7-grams. Second, we hope to optimize our segmentation threshold α for each language separately via co-training of ParaMor against Morfessor. We are also interested in using statistical models to simulate rule-based and black-box systems from other areas of NLP.

Acknowledgements. This research was supported in part by NSF Grant #IIS-0811745 and DOD/NGIA grant #HM1582-08-1-0038. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF or DOD.

References

1. Oflazer, K., El-Kahlout, İ.D.: Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation. In: Statistical MT Workshop at ACL (2007)
2. Creutz, M.: Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition. Ph.D. Thesis, Computer and Information Science, Report D13, Helsinki, University of Technology, Espoo, Finland (2006)
3. Harris, Z.: From Phoneme to Morpheme. Language, 31(2), 190-222 (1955); Reprinted in Harris, Z.: Papers in Structural and Transformational Linguists. Reidel D. (ed.), Dordrecht (1970)
4. Bernhard, D.: Simple Morpheme Labeling in Unsupervised Morpheme Analysis. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 873–880. Springer, Heidelberg (2008)
5. Goldsmith, J.: Unsupervised Learning of the Morphology of a Natural Language. Computational Linguistics 27(2), 153–198 (2001)
6. Snyder, B., Barzilay, R.: Unsupervised Multilingual Learning for Morphological Segmentation. In: Proceedings of ACL 2008: HLT (2008)
7. Poon, H., Cherry, C., Toutanova, K.: Unsupervised Morphological Segmentation with Log-Linear Models. In: Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL (2009)
8. Monson, C.: ParaMor: From Paradigm Structure to Natural Language Morphology Induction. Ph.D. Thesis, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania (2009)
9. Monson, C., Carbonell, J., Lavie, A., Levin, L.: ParaMor and Morpho Challenge 2008. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G.J.F., Kurimo, M., Mandl, T., Peñas, A., Petras, V. (eds.) CLEF 2008. LNCS, vol. 5706, pp. 967–974. Springer, Heidelberg (2009)
10. Tjong Kim Sang, E. F.: Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of CoNLL 2002 (2002)
11. Tjong Kim Sang, E. F., Buchholz, S.: Introduction to the CoNLL-2000 Shared Task: Chunking. In: Computational Natural Language Learning, CoNLL (2000)
12. Roark, B., Hollingshead, K.: Linear Complexity Context-Free Parsing Pipelines via Chart Constraints. In: Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL (2009)
13. Xue, N.: Chinese Word Segmentation as Character Tagging. Computational Linguistics and Chinese Language Processing 8(1), 29–47 (2003)
14. Hollingshead, K., Fisher, S., Roark, B.: Comparing and Combining Finite-State and Context-Free Parsers. In: Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP (2005)
15. Collins, M.: Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In: Conference on Empirical Methods in Natural Language Processing, EMNLP (2002)
16. Kurimo, M., Virpioja, S., Turunen, V.T., Blackwood, G.W., Byrne, W.: Overview and Results of Morpho Challenge 2009. In: 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, Revised Selected Papers. LNCS, Springer, Heidelberg (2010)

17. Trón, V., Gyepesi, G., Halácsy, P., Kornai, A., Németh, L., Varga, D.: Hunmorph: Open Source Word Analysis. In: ACL Workshop on Software (2005)
18. Varga, D., Halácsy, P., Kornai, A., Németh, L., Trón, V., Váradi, T., Sass, B., Bottyán, G., Héja, E., Gyarmati, Á., Mészáros, Á., Labundy, D.: Hunglish corpus, http://mokk.bme.hu/resources/hunglishcorpus (accessed on August 18, 2009)

A Rule-Based Acquisition Model Adapted for Morphological Analysis*

Constantine Lignos[1], Erwin Chan[2], Mitchell P. Marcus[1], and Charles Yang[1]

[1] University of Pennsylvania
lignos@cis.upenn.edu, mitch@cis.upenn.edu, charles.yang@ling.upenn.edu
[2] University of Arizona
echan3@u.arizona.edu

Abstract. We adapt the cognitively-oriented morphology acquisition model proposed in (Chan 2008) to perform morphological analysis, extending its concept of base-derived relationships to allow multi-step derivations and adding features required for robustness on noisy corpora. This results in a rule-based morphological analyzer which attains an F-score of 58.48% in English and 33.61% in German in the Morpho Challenge 2009 Competition 1 evaluation. The learner's performance shows that acquisition models can effectively be used in text-processing tasks traditionally dominated by statistical approaches.

1 Introduction

Although extensive work has been done on creating high-performance unsupervised or minimally supervised morphological analzyers (Creutz and Lagus 2005, Monson 2008, Wicentowski 2002), little work has been done to bridge the gap between the computational task of morphological analysis and the cognitive task of morphological acquisition. We address this by adapting the acquisition model presented in (Chan 2008) to the task of morphological analysis, demonstrating the effectiveness of cognitively-oriented models on analysis tasks.

The most well-known cognitive models (Pinker 1999, Rumelhart and McClelland 1986) are poorly suited for unsupervised morphological analysis given that they are commonly focused on a single morphological task, the English past tense, and are based on the assumption that pairs of morphologically related words, such as *make/made*, are given to the learner. While there is evidence that clustering-based approaches can identify sets of morphologically related words (Parkes et al. 1998, Wicentowski 2002), word-pair based algorithms have only been evaluated on error-free pairs.

Many computational models have focused on segmentation-based approaches, most commonly using simple transitional-probability heuristics (Harris 1955,

* Thanks to Jana Beck for her assistance in analyzing the German results and for her insightful comments throughout the development process. Portions of this paper were adapted from the material presented in the CLEF 2009 Morpho Challenge Workshop (Lignos et al. 2009).

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 658–665, 2010.

1. Pre-process words and populate the Unmodeled set.
2. Until a stopping condition is met, perform the main learning loop:
   (a) Count suffixes in words of the Base ∪ Unmodeled set and the Unmodeled set.
   (b) Hypothesize transforms from words in Base ∪ Unmodeled to words in Unmodeled.
   (c) Select the best transform.
   (d) Reevaluate the words that the selected transform applies to, using the Base, Derived and Unmodeled sets
   (e) Move the words used in the transform accordingly.
3. Break compound words in the Base and Unmodeled sets.

Fig. 1. The Learning Algorithm

Keshava and Pitler [2006]), or n-gram-based statistical models (most recently Spiegler [2009]). Often segmentation-based approaches organize the segmentations learned into paradigms (Goldsmith [2001], Monson [2008]). While the use of paradigms creates what appears to be a useful organization of the learned rules, recent work questions the learnability of paradigms from realistic input (Chan [2008]).

Although the highest performance has traditionally come from segmentation-based approaches, it is difficult to define linguistically reasonable segmentation behavior for even simple cases (*make/mak + ing*), and from the point of view of an acquisition model segmentation suggests a notion of an abstract stem whose psychological and linguistic reality is not obvious (Halle and Marantz, 1993).

This research seeks to build a practical morphological analyzer by adapting a cognitive model that embraces the sparsity seen among morphological forms and learns a linguistically inspired representation. By doing so, we bring computational and cognitive models of morphology learning closer together.

2 Methodology

We use the Base and Transforms Model developed in (Chan, 2008 chap. 5) and extend the accompanying algorithm to create a morphological analyzer. We present a brief summary of the Base and Transforms model here and present our modified version of the algorithm. Our algorithm is summarized in Figure 1.

2.1 The Base and Transforms Model

A morphologically derived word is modeled as a base word with an accompanying transform that changes the base to create a derived form. A base must be a word observed in the input, not an abstract stem, and a transform is an orthographic modification made to a base to create a derived form. It is defined as two affixes $(s1, s2)$, where $s1$ is removed from the base before concatenating $s2$. Thus to

derive *making* from *make* we apply the transform (e, ing), removing $-e$ from the base and then concatenating $-ing$. We represent a null suffix as \$. A transform also has a corresponding word set, which is the set of base-derived pairs that the transform accounts for. The bases of a transform are the only words that the transform can be applied to.

We now give an overview here of the learning algorithm used in this work. For further details on the algorithm's implementation and performance, see (Lignos et al., 2009).

Word Sets. Each word in the corpus belongs to one of three word sets at any point in execution: Base, Derived, or Unmodeled. The Base set contains the words that are used as bases of learned transforms but are not derived from any other form. The derived set contains words that are derived forms of learned transforms, which can also serve as bases for other derived forms. All words begin in the Unmodeled set and are moved into Base or Derived as transforms are learned.

Pre-processing. We perform a minimal amount of pre-processing to support learning on hyphenated words. Any word with a hyphen is placed into a set of words excluded from the learning process, but each segment in the hyphenated word is included in learning. For example, *punk-rock-worshipping* would not be included in learning, but *punk*, *rock*, and *worshipping* would. The analysis of any hyphenated word is the concatenation of the analysis of its segments, in this case *PUNK ROCK WORSHIP +(ing)*.

2.2 The Learning Loop

Affix Ranking. We count the affixes contained in each word in the base and unmodeled sets by brute force, scanning the first and last 5 letters in each word. To prevent rare words and foreign words from affecting the affix and transform ranking process, words only count toward an affix or transform's score if they are relatively frequent in the corpus. For a word to be considered common, it must appear more than once in the corpus and have a frequency greater than one in one million. This frequency cutoff was set by examining the list of words in the Morpho Challenge 2009 evaluation corpora above the cutoff frequency to find a point where less common morphological productions are still included but most typos and foreign words are excluded.

Transform Ranking. We hypothesize transforms of all combinations of the top 50 affixes and count the number of base-derived pairs in each transform. The score of a transform is the number of word pairs it accounts for multiplied by the net number of characters that the transform adds or removes to a base. For example, if the transform $(e,\ ing)$, which removes one letter from the base and adds three, has 50 base-derived pairs, its score would be $50 * |3 - 1| = 100$.

To approximate orthographic gemination and the merging of repeated characters when a morpheme is attached, we relax the conditions of testing whether a

base-derived pair is acceptable. For each potential base word for a transform, we compute two derived forms: a standard derived form that is the results of applying the transform precisely to the base, and a "doubled" derived form where $s1$ is removed from the base, the last character of the remaining base is repeated, and then $s2$ is attached. For example, when checking the transform ($, *ing*) applied to *run*, we generate the standard derived form *runing* and the doubled form *running*. Additionally, in cases where the final character of the base after $s1$ has been removed is the same as the first character of $s2$, we also create an "undoubled" derived form where the first character of $s2$ is removed such that applying the transform does not result in a repeated character. For example, when applying ($, *ed*) to *bake*, the standard form would be *bakeed*, but the undoubled form would be *baked*. All derived forms that are observed in the Unmodeled set are added, so if the standard, doubled, and undoubled forms are all observed, three base-derived pairs would be added to the transform. These doubling and undoubling effects are most commonly attested in English, but the doubling and undoubling rules are designed to allow the learner to broadly approximate orthographic changes that can occur when morphemes are attached in any language.

Transform Selection. The learner selects the transform of the higest rank that has acceptable segmentation precision. Segmentation precision represents the probability that given any Unmodeled word containing $s2$ reversing the transform in question will result in a word. Segmentation precision must exceed a set threshold for the learner to accept a hypothesized transform. By observing the precision of transforms during development against the Brown corpus, we set a threshold of 1% as the threshold of an acceptable transform. If more than 20 transforms are rejected in an iteration because of unacceptable segmentation precision, the learning loop stops as it is unlikely that there are good transforms left to model.

Transform Word Set Selection. After a transform is selected, we apply the selected transform as broadly as possible by relaxing word sets that the transform's bases and derived words can be members of. This allows our algorithm to handle multi-step derivations, for example to model derivational affixes on an base that is already inflected or allow derived forms to serve as bases for unmodeled words.

This expansion of the permissible types of base/derived pairs requires similar changes to how words are moved between sets once a transform has been selected. We developed the following logic for moving words:

1. No word in Base may be the derived form of another word. If a word pair of the form Base → Base is used in the selected transform, the derived word of that pair is moved to Derived. After movement the relationship is of the form Base → Derived.
2. A word in Derived may be the base of another word in Derived. If a word pair of the form Derived → Unmodeled is used in the selected transform, the derived word of that pair is moved to Derived, and the base word remains in Derived. After movement the relationship is of the form Derived → Derived.

Table 1. Transforms learned in English and German on Morpho Challenge 2009 evaluation data sets

English		
	Trans.	Sample Pair
1	+($, s)	scream/screams
2	+($, ed)	splash/splashed
3	+($, ing)	bond/bonding
4	+($, 's)	office/office's
5	+($, ly)	unlawful/unlawfully
6	+(e, ing)	supervise/supervising
7	+(y, ies)	fishery/fisheries
8	+($, es)	skirmish/skirmishes
9	+($, er)	truck/trucker
10	($, un)+	popular/unpopular
11	+($, y)	risk/risky
12	($, dis)+	credit/discredit
13	($, in)+	appropriate/inappropriate
14	+($, ation)	transform/transformation
15	+($, al)	intention/intentional
16	+(e, tion)	deteriorate/deterioration
17	+(e, ation)	normalize/normalization
18	+(e, y)	subtle/subtly
19	+($, st)	safe/safest
20	($, pre)+	school/preschool
21	+($, ment)	establish/establishment
22	($, inter)+	group/intergroup
23	+(t, ce)	evident/evidence
24	($, se)+	cede/secede
25	+($, a)	helen/helena
26	+(n, st)	lighten/lightest
27	($, be)+	came/became

German		
	Trans.	Sample Pair
1	+($, en)	produktion/produktionen
2	+($, er)	ueberragend/ueberragender
3	+($, es)	einfluss/einflusses
4	+($, s)	gewissen/gewissens
5	+($, ern)	schild/schildern
6	+(r, ern)	klaeger/klaegern
7	($, ver)+	lagerung/verlagerung
8	($, ge)+	fluegel/gefluegel
9	($, ueber)+	nahm/uebernahm
10	($, vor)+	dringlich/vordringlich
11	($, be)+	dachte/bedachte
12	($, unter)+	schaetzt/unterschaetzt
13	($, ein)+	spruch/ einspruch
14	($, er)+	sucht/ersucht
15	($, auf)+	ruf/aufruf
16	($, an)+	treibt/antreibt
17	($, zu)+	teilung/zuteilung
18	($, aus)+	spricht/ausspricht
19	($, ab)+	bruch/abbruch
20	($, ent)+	brannte/entbrannte
21	($, in)+	formiert/informiert
22	+(t, ren)	posiert/posieren
23	+($, lich)	dienst/dienstlich
24	($, un)+	wichtig/unwichtig
25	+(t, rung)	rekrutiert/rekrutierung
26	($, he)+	rauf/herauf

2.3 Post-processing

Once the learning loop has stopped, the learner tries to break the compound words that remain in the Base and Unmodeled sets using a simple 4-gram character-level model trained on the words in Base. Words are broken at the lowest point of forward probability if the resulting substrings are words seen in the input. For further detail, see (Lignos et al. 2009).

3 Results

3.1 Performance

The learner completes 27 iterations in English and 26 iterations in German before stopping. The resulting analyses achieve an F-measure of 58.48% in English and 33.61% in German in the official Morpho Challenge 2009 competition 1 evaluation, learning the rules presented in Table 1. Among non-baseline methods in competition 1, a comparison against a linguistic gold standard, it achieved the

third highest F-measure and highest precision in English, and the 11th highest F-measure and highest precision in German. Among non-baseline methods in competition 2, an information retrieval task, it achieved the highest average precision in English and the 7th highest in German.

3.2 Errors

While it is difficult to assign precise, mutually exclusive categories to the learner's errors, they can be grouped into these categories:

Rare affixes. Many productive affixes in the gold standard are rarer than would be expected in the training corpus, for example the English suffixes $-ness$ and $-able$, and thus the learner fails to distinguish them from noise in the data.

Unproductive affixes. Some affixes in the gold standard are no longer productive in the language being learned. For example, the gold standard suggests that *embark* be analyzed as $em + bark$, but the Germanic prefix $em-$ is not productive in modern English and thus appears in few pairs. It is unlikely that a cognitively oriented learner would learn these rules from the input data.

Multi-step derivations. The learner fails to learn multi-step derivations, for example *acidified* as *ACID +ify +ed*, if any intermediate derivations (*acidify*) are not present in the corpus. These multi-step derivations account for the lower recall of the learner compared to other methods in Morpho Challenge 2009. However, the absence of errors in attempting to generalize rules to analyze these derivations is partly responsible for the learner's high precision.

Spurious relationships. The learner can form word pairs of unrelated words that fit the pattern of common rules, for example *pin/pining* in English. In German, this appears to cause a significant number of errors for even very frequent transforms. In the development set, the three most common transforms in German have a precision of 47.4%, while in English they have a precision of 83.9%.

4 Discussion

4.1 Limitations of the Algorithm

By learning individual transforms rather than full paradigms, the learner avoids a major consequence of sparsity in morphology learning. However, the algorithm must observe all steps of a multi-step derivation to learn the connection between the words in the derivation. This limitation has little impact in English, but in languages with more morphemes per word, such as German, this is a limiting factor in the algorithm's performance. With a larger number of morphemes per

word, it is unlikely that all permutations of the morphemes would occur with the same base. Segmentation-based approaches have a natural advantage in this area. They need only identify the morphemes and decide whether to apply them to an individual word, unlike our algorithm which identifies rules but requires a minimal pair of words that show a rule's applicability.

While the learner's current approach results in very high precision, it does not match the kind of rule generalization desirable for an acquisition model and results in poorer performance when there are many morphemes per word. In order to address this, the learner must understand the conditions for applying rules. This will require unsupervised part of speech induction so that rules can be marked as inflectional or derivational and using POS to decide whether a rule should be applied. A POS-aware version of the algorithm would likely achieve higher precision as it would not pair words of inappropriate POS together for a given transform. The ability to generalize in this fashion would enable the learner to analyze unseen words, which the learner cannot currently do.

4.2 Limitations of the Rule Representation

The simple definition of a rule as an affix-change operation limits the languages that the learner can currently be applied to. Languages with vowel harmony, such as Finnish and Turkish, require a more complex and phonologically-specified representation to be accurately modeled using a rule-based approach. Languages that use non-concatenative morphology, such as Arabic and Hebrew, cannot be modeled in any meaningful way using our rule representation, as the algorithm only searches for affix changes and not word-medial changes.

These shortcomings are not inherent to the Base and Transforms model but rather specific to the transform representation used. Expanding the transform definition to support infixes would be a first step to supporting nonconcatenative languages, but operations like vowel harmony and stem changes require a level of phonological information that has thus far not been used in unsupervised morphological analyzers. A more likely approach to handling vowel harmony may be to merge morphemes that appear in similar contexts (Can, 2009).

4.3 Conclusions

The high performance of the learner in English and German suggests that an acquisition model can perform at a comparable level to statistical models. Future work should focus on the expansion of acquisition models to support a richer set of morphological phenomena and finer-grained representation of the morphological rules learned.

References

Brent, M.R., Murthy, S.K., Lundberg, A.: Discovering morphemic suffixes: A case study in minimum description length induction. In: Proceedings of the Fifth International Workshop on AI and Statistics (1995)

Can, B., Manandhar, S.: Unsupervised Learning of Morphology by Using Syntactic Categories. In: Working Notes of the 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30–October 2 (2009)

Chan, E.: Structures and distributions in morphology learning. PhD Thesis, University of Pennsylvania (2008)

Creutz, M., Lagus, K.: Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0. Publications in Computer and Information Science, Report A81, Helsinki University of Technology (March 2005)

Goldsmith, J.: Unsupervised learning of the morphology of a natural language. Computational Linguistics 27(2), 153–198 (2001)

Halle, M., Marantz, A.: Distributed morphology and the pieces of inflection. The view from Building 20, 111–176 (1993)

Harris, Z.S.: From phoneme to morpheme. Language, 190–222 (1955)

Keshava, S., Pitler, E.: A simpler, intuitive approach to morpheme induction. In: Proceedings of 2nd Pascal Challenges Workshop, pp. 31–35 (2006)

Lignos, C., Chan, E., Marcus, M.P., Yang, C.: A Rule-Based Unsupervised Morphology Learning Framework. In: Working Notes of the 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30–October 2 (2009)

Monson, C.: ParaMor: from Paradigm Structure to Natural Language Morphology Induction. PhD Thesis, Carnegie Mellon University

Parkes, C.H., Malek, A.M., Marcus, M.P.: Towards Unsupervised Extraction of Verb Paradigms from Large Corpora. In: Proceedings of the Sixth Workshop on Very Large Corpora, Montreal, Quebec, Canda, August 15-16 (1998)

Pinker, S.: Words and rules: The ingredients of language. Basic Books, New York (1999)

Rumelhart, D.E., McClelland, J.L.: Parallel distributed processing: Explorations in the microstructure of cognition. Psychological and biological models, vol. 2. MIT Press, Cambridge (1986)

Spiegler, S., Golnia, B., Flach, P.: PROMODES: A probabilistic generative model for word decomposition. In: Working Notes of the 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, September 30–October 2 (2009)

Wicentowski, R.: Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework. Ph. D. thesis, Johns Hopkins University (2002)

Morphological Analysis by Multiple Sequence Alignment

Tzvetan Tchoukalov[1], Christian Monson[2], and Brian Roark[2]

[1] Stanford University
[2] Center for Spoken Language Understanding, Oregon Health & Science University
eurus@stanford.edu, monsonc@csee.ogi.edu, roark@cslu.ogi.edu

Abstract. In biological sequence processing, Multiple Sequence Alignment (MSA) techniques capture information about long-distance dependencies and the three-dimensional structure of protein and nucleotide sequences without resorting to polynomial complexity context-free models. But MSA techniques have rarely been used in natural language (NL) processing, and never for NL morphology induction. Our MetaMorph algorithm is a first attempt at leveraging MSA techniques to induce NL morphology in an unsupervised fashion. Given a text corpus in any language, MetaMorph sequentially aligns words of the corpus to form an MSA and then segments the MSA to produce morphological analyses. Over corpora that contain millions of unique word types, MetaMorph identifies morphemes at an F_1 below state-of-the-art performance. But when restricted to smaller sets of orthographically related words, MetaMorph outperforms the state-of-the-art ParaMor-Morfessor Union morphology induction system. Tested on 5,000 orthographically similar Hungarian word types, MetaMorph reaches 54.1% and ParaMor-Morfessor just 41.9%. Hence, we conclude that MSA is a promising algorithm for unsupervised morphology induction. Future research directions are discussed.

1 Introduction

Biologists are interested in the function of genes and proteins. Since organisms evolve by the slow mutation of individual base pairs in their DNA, gene regions from related organisms that consist of similar sequences of nucleotides will likely perform similar functions. Multiple Sequence Alignment (MSA) techniques are one suite of tools that computational biologists use to discover nucleotide sequences that are unusually similar, and that thus likely serve similar biological functions [1].

Like biologists, linguists are interested in the sub-regions of longer linear sequences that serve particular functions. Where biologists look at strings of nucleotide bases in DNA or sequences of amino acids in proteins, linguists examine the strings of phonemes or written characters that form words. And where biologists seek genes, linguistics identify morphemes—the smallest linguistic units that carry meaning. Given the similarities between biological and linguistic sequences, we seek to transfer the successes of MSA models from biology to induce natural language morphology in an unsupervised fashion.

C. Peters et al. (Eds.): CLEF 2009 Workshop, Part I, LNCS 6241, pp. 666–673, 2010.

Although we are inspired by biology, building MSAs for natural language morphology induction is fundamentally different from building MSAs in biological applications. In biology, it is typical to align a few sequences (on the order of 10) of very long length (perhaps millions of base pairs). In our NL morphology application, the sequences are words and are thus relatively short (on the order of 10 characters)—but there may be tens of thousands or even millions of distinct word types to align. Moreover, our goals are somewhat different from the goals that biologists typically have when applying MSA techniques. We wish to definitively segment words into separate morphemes, but we have not encountered any work in computational biology that uses MSA to segment out genes. Instead, biologists use MSA to merely identify regions of likely similarity between sequences.

Relating our MSA work to research directions in NL morphological processing: While we are unaware of any prior attempt to model the structure of NL morphology using MSA techniques, MSA is at base a method for measuring distances between strings—and string edit distances *have* played a part in a variety of unsupervised morphology induction systems. Baroni et al. [2], for example, seed a semantically based induction system with pairs of words that are orthographically similar. Likewise, Wicentowski [3] trains a statistical model of morphological structure from several weak correlates of morphological relatedness—including the Levenshtein distance between pairs of words. Readers interested in unsupervised morphology induction more broadly may consult Chapter 2 of [4].

2 The MetaMorph Multiple Sequence Alignment Algorithm

The input to an MSA algorithm is a set of sequences; and the output is an alignment of the elements of the sequences. Fig. 1 depicts an alignment over a set of ten English words. Each sequence, i.e. each word, in Fig. 1 forms a separate row of the alignment table. An MSA algorithm places the characters of each sequence into aligned columns. The order of elements in each sequence is fixed, but, to improve an alignment, an MSA algorithm may place gaps, '-', in some columns between the characters of a sequence. The ten sequences of Fig. 1 are arranged into eight aligned columns.

A variety of algorithms could produce a multiple sequence alignment like that in Fig. 1. Our MetaMorph algorithm employs two standard MSA algorithms in turn: *Progressive Alignment* [5] and a *Profile Hidden Markov Model (HMM)* [1]. Both Progressive Alignment and Profile HMMs define position specific distributions over characters. Fig. 2 displays the position specific character distributions for the alignment in Fig. 1. For each of the eight columns of the alignment table, Fig. 2 has a corresponding column that contains a smoothed probability distribution over the alphabet of characters that appears in Fig. 1: Each column distribution in Fig. 2 contains a count for each occurrence of each character in the corresponding column of Fig. 1, plus a Laplace smoothing constant of one for each character. We treat the gap as simply another character of the alphabet. The probability of a character given a column is the ratio of the character count in that column's distribution to the distribution total. For example, in Figs. 1 and 2, the probability of the character 'd' given column 1 is 5/26.

```
12345678
d-anc-es
d-anc-ed
d-anc-e-
d-ancing
r-unning
j-umping
j-ump-ed
j-ump-s-
j-ump---
laughing
```

Fig. 1. A sample multiple sequence alignment (MSA)

Chars	1	2	3	4	5	6	7	8
a	1	2	5	1	1	1	5	1
c	1	1	1	1	5	1	1	1
d	5	1	1	1	1	1	1	3
e	1	1	1	1	1	1	1	1
g	1	1	1	2	1	1	1	5
h	1	1	1	1	2	1	1	1
i	1	1	1	1	1	5	1	1
j	5	1	1	1	1	1	1	1
l	2	1	1	1	1	1	1	1
m	1	1	1	5	1	1	1	1
n	1	1	1	6	2	1	5	1
p	1	1	1	1	5	1	1	1
r	2	1	1	1	1	1	1	1
s	1	1	1	1	1	1	2	2
u	1	1	7	1	1	1	1	1
gap	1	10	1	1	1	7	2	4

Fig. 2. Laplace-smoothed (count plus 1) position specific character distributions for the MSA in Fig. 1

2.1 Building an Initial MSA via Progressive Alignment

MetaMorph begins with a progressive alignment algorithm that builds an initial alignment over an orthographically similar subset of the full input corpus. Progressive alignment algorithms build an alignment for a set of sequences iteratively. After a first pair of sequences are aligned to each other, a third sequence is aligned to the newly formed alignment; a fourth sequence follows; and a fifth, etc. The size of the orthographically similar subset of corpus words is a free parameter. We experimented with subsets that contained between 5,000 and 20,000 words.

Step 1: Ordering. Before MetaMorph aligns words, our algorithm orders the words which will form the initial MSA. The first two words in the ordered list are the Levenshtein most similar pair of words from the 1000 most frequent words of the input corpus. MetaMorph then sequentially adds words to our ordered list by identifying, from all the words in the input corpus, the word that is most similar to some word already in the ordered list. To ensure that the initial MSA contains standard natural language words, we require all words in the ordered list to be between 5 and 16 characters in length and to contain no hyphens or numbers. MetaMorph continues to add words to the ordered list until a preset size limit is reached.

Step 2: Alignment. To produce an MSA from the ordered list of words, MetaMorph initializes an MSA to the first word in the sorted list. Each character in the first word appears in a separate column. Using Laplace smoothing, MetaMorph then calculates the (trivial) column distributions over the characters of the alphabet, a la Fig. 2.

For each remaining word, *w*, in the ordered list, MetaMorph uses a dynamic programming algorithm to, in turn, identify the lowest-cost alignment of *w* to the current MSA. Beginning with the first character of *w* and the first column of the MSA, there are three possible alignment choices: First, the character may be aligned to the column; Second, the column may be aligned to a gap that is inserted into *w*; and third, the character may be aligned to a column of gaps that is inserted into the MSA. We

define the cost of matching a character, *c*, to an MSA column, *l*, to be the negative logarithm of the probability of *c* occurring in *l*. Treating gaps as just another character, the cost of aligning the column *l*, to a gap in the word *w* is simply the negative logarithm of the probability of a gap in *l*. And we measure the cost of matching a character, *c*, to a newly-inserted column of gaps as the negative logarithm of the (smoothed) probability of *c* appearing in a column that thus far contains only gaps. As the number of words in the alignment increases, the contribution of Laplace smoothing to the overall character distribution decreases, and hence the cost of inserting a column of gaps into the MSA increases.

The score of an alignment of a word, *w*, to an MSA is the sum of the match costs and gap insertions costs specified by that particular alignment. Dynamic programming with back-tracing finds the optimal alignment of each *w* in O(NM) time, where N is the length of *w*, and M the length of the current MSA. When all words in the ordered wordlist have been inserted into the MSA, the initial alignment cycle is complete.

Step 3: Realignment. After the initial alignment phase, MetaMorph performs leave-one-out refinement [6]: Sequentially, MetaMorph removes each word from the MSA, and then realigns the removed word to the remaining MSA. MetaMorph halts leave-one-out refinement when one of two criteria is met: 1. The MSA remains unchanged, or 2. The sum of the entropies of the column distributions increases after a set number of realignment cycles. Leave-one-out refinement is designed to specialize each column on a smaller selection of characters. If entropy rises, then the columns are permitting a wider variety of characters, and we halt realignment.

2.2 Finalizing Alignment via a Profile HMM

Once MetaMorph's progressive alignment phase has built an alignment over an orthographically similar subset of the full corpus, our algorithm freezes the initial alignment as a Profile HMM. Each column of the progressively built alignment acts as an HMM state whose character production probabilities correspond to the column's character distribution. Each word of the corpus that was not in the original orthographically similar subset is then aligned to the Profile HMM.

2.3 Segmentation

Having obtained a final MSA, we must now produce a morphological segmentation of the input corpus. To motivate the segmentation strategy that the MetaMorph algorithm employs, Fig. 3 depicts six sequences of an MSA built over a Hungarian corpus of 500,000 words. The first five sequences in Fig. 3 are legitimate inflections of the word in the first sequence: 'között' is a postposition in Hungarian, while 'i', 't', 'e', and 'em' are suffixes that attach to postpositions. In contrast, the last sequence in Fig. 3, 'kötöttem' is an inflected verb that is linguistically unrelated to the other sequences. A reasonable segmentation of the final sequence would be 'köt-ött-em', with 'köt' a verb stem meaning 'tie', 'ött' marking past tense, and 'em' 1st person singular.

To segment an MSA into morphemes, our MetaMorph algorithm selects a set of columns in the MSA and segments *all* the words of the corpus at those columns. Examining Fig. 3, the pattern of gap columns does not appear to indicate where a morpheme boundary should be placed: columns of gaps separate *all* the characters of all

```
-----k----ö---z-----ö-------t-----------t---------
-----k----ö---z-----ö-------t-----------t----i----
-----k----ö---z-----ö-------t-----------t----i-t--
-----k----ö---z-----ö-------t-----------t----e----
-----k----ö---z-----ö-------t-----------t----e-m--
-----k----ö---t-----ö-------t-----------t----e-m--
```

Fig. 3. Six Sequences from an MSA induced over a Hungarian corpus

the words, with long sequences of gap columns sometimes occurring internal to a morpheme (eleven gaps separate the doubled 't's for example). Other obvious techniques, such as segmenting at columns with the maximal gap probability or at those columns whose probability distributions had minimal entropy, generated completely implausible segmentations.

Lacking a segmentation strategy that relies solely on the MSA, MetaMorph instead leverages knowledge from an independent algorithm for unsupervised induction of morphology. The independent system which we used is the ParaMor-Morfessor Union system from Morpho Challenge 2009 [7]. The ParaMor-Morfessor system placed first at F_1 for morpheme identification in three of the five languages of Morpho Challenge 2009. To segment the words of a corpus, the MetaMorph algorithm searches for a set of segmentation columns that maximize the F_1 score *against the independent (still unsupervised) system*.

MetaMorph uses a greedy search algorithm to decide upon a set of segmentation columns. One at a time, MetaMorph considers each column in the MSA as a potential segmentation point. The segmentations that result from a particular column are scored against the analyses provided by the independent morphology segmentation algorithm. MetaMorph retains that segmentation column which most improves F_1, and then iteratively considers adding a second segmentation column. If, after any iteration, no column is found to improve F_1, MetaMorph terminates the search for additional segmentation columns. Although the segmentation columns are fixed for all words, the final number of morphemes in any particular word will still vary because, after segmenting a word, MetaMorph discards morphemes that consist solely of gaps.

3 Results and Conclusions

To evaluate the success of the MetaMorph algorithm, we participated in Morpho Challenge 2009 [8], where ten groups from around the world assessed their unsupervised morphology induction systems with both linguistic and task-based criteria. The linguistic evaluation of Morpho Challenge measured the precision, recall, and F_1 score of each unsupervised algorithm at identifying the constituent morphemes of the words in a corpus. Of the six tracks of the linguistic competition, MetaMorph had the least success at the two Arabic scenarios and the most success on Turkish. MetaMorph's poor performance on Arabic, less than 6% F_1 for both the vowelized and unvowelized tracks, is directly attributable to MetaMorph's reliance on the ParaMor-Morfessor Union system for its segmentation strategy. The Union system also suffered its poorest performance on the Arabic tracks, F_1 scores of less than 10%.

In Turkish, MetaMorph outperformed at F_1 the baseline unsupervised system of Morpho Challenge, a system named Morfessor [9]: MetaMorph achieved an F_1 score of 33.6% where Morfessor came in at 29.7%. Backstopping MetaMorph's comparatively strong performance on Turkish is the highest absolute recall score that MetaMorph attained for any language, 29.5%. But since absolute scores of morpheme identification are not comparable across languages, consider MetaMorph's recall as a fraction of the recall score (60.4%) of that system [7] which had the highest F_1 score at Morpho Challenge for Turkish: this recall fraction is 0.488 (i.e. 29.5% / 60.4%). If we calculate MetaMorph's recall fraction against the F_1-best system for the other non-Arabic languages of Morpho Challenge we get 0.401 for English, but just 0.322 for German and 0.300 for Finnish, Interestingly, there are many fewer word types in the Turkish and English data sets (617,000 and 385,000 respectively) of Morpho Challenge than there are in the Finnish and German sets (2.21 and 1.27 million respectively). The following experiments suggest, counterintuitively, that it may be the smaller size of the Turkish and English data sets that lead to MetaMorph's higher recall.

We used Hunmorph [10], a hand-built Hungarian morphological analyzer, to produce a morphological answer key containing 500,000 unique Hungarian word types from the Hunglish corpus [11]. Over the full Hungarian corpus, MetaMorph's F_1 score reached a paltry 19.7%. But we found that if we restricted our evaluation to just those words from which MetaMorph's progressive alignment algorithm constructed the initial MSA, performance improved dramatically.

We ran three experiments. In the first experiment, MetaMorph's progressive alignment algorithm built an MSA from a set of 5,000 orthographically similar words, i.e., the size limit in step 1 of Section 2.1 is set to 5,000; in the second, MetaMorph used a size limit of 10,000 words; and in the third, a size limit of 20,000. During the segmentation phase of these three experiments, we instructed MetaMorph's greedy search to select segmentation columns that maximized F_1 score, *not* over the full Hungarian corpus, but rather over just the words in the smaller set of orthographically similar words. Using the Linguistic evaluation procedure from Morpho Challenge, we then evaluated MetaMorph's morphological analyses over these same three (smaller) sets of orthographically similar words. Additionally, we evaluated the performance of the ParaMor-Morfessor Union system over these same three sets of words. To evaluate the Union system we used the full Hungarian corpus to induce segmentations, but restricted our evaluations to the sets of 5,000, 10,000, and 20,000 words. The results of these experiments appear in Fig. 4.

Immediately striking from Fig. 4 is that MetaMorph significantly outscores the ParaMor-Morfessor Union over the 5,000 and 10,000 word sets. Remember that MetaMorph's segmentation phase seeks to emulate the segmentations that the ParaMor-Morfessor Union system produces. Over small datasets, MetaMorph has successfully generalized beyond the system that is used as a segmentation guide.

Furthermore, as the set size increases in Fig. 4, MetaMorph's F_1 steadily decreases. Since each of our three experiments is learning from and evaluating over a different set of words, one explanation for MetaMorph's downward trend might simply be that the set of 20,000 words contained more inherent morphological complexity than the smaller sets. But this explanation fails when we examine the performance of the ParaMor-Morfessor Union system over these same data sets: The Union system's

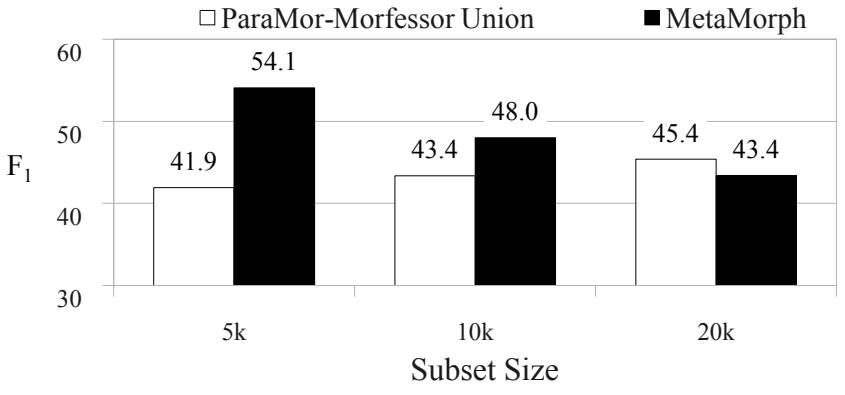


Fig. 4. The F_1 performance of two unsupervised morphology induction algorithms for three subsets of a Hungarian corpus

performance increases in step with the set size. Instead, MetaMorph's strong performance at the smallest word-set sizes, leads us to conclude that MSA is most effective when used over a smaller set of words that exhibit orthographic similarity. This same effect may be at work in MetaMorph's higher recall scores for English and Turkish in Morpho Challenge proper.

For some tasks, such as machine translation (MT), MetaMorph's conservative lower-recall approach to morphological segmentation pays off: MetaMorph took second place in Finnish MT at Morpho Challenge 2009, with a BLEU score of 28.20. In the MT evaluation the words of a non-English language text were replaced by their automatic morphological analyses before applying a statistical MT algorithm. The baseline statistical MT system that translates directly from words had a BLEU score of 27.64; thus MetaMorph improves BLEU score over the word-based model by 0.56 BLEU points. For exhaustive details about the absolute and relative performance of the MetaMorph algorithm at Morpho Challenge 2009, see [8].

The Next Steps. MetaMorph's success at analyzing the morphological structure of smaller, more focused, sets of words suggests that in future we use progressive alignment techniques to build a number of separate alignment structures focused on different subsets of the full corpus. Each subset of the corpus would contain orthographically similar words. It may also be the case that the optimal number of words for which to build an alignment is smaller than 5,000. Indeed, the optimal set size may vary by language, or even by part-of-speech within a language.

A second weakness that we would like to address in the MetaMorph algorithm is the poor correlation between the arrangement of columns in the MSA and the arrangement of morphemes within words. Where most morphemes are contiguous sequences of characters, MetaMorph's MSA columns place gap symbols internal to true morphemes. Better correlation between MSA columns and morphemes in words would free MetaMorph from relying on an independent unsupervised morphology induction system during segmentation (Section 2.3). A match cost function that accounted for string position is one potential method for producing contiguous morpheme columns. We might, for example, lower match costs (and thereby reduce the

number of gap columns) near the middle of an MSA to allow for the wide variety of characters that are likely in the stems of words. Another possible solution may be to tie the character distributions of neighboring MSA columns so as to avoid overtraining a column to a particular character.

Acknowledgements. This research was supported in part by NSF Grant #IIS-0811745 and DOD/NGIA grant #HM1582-08-1-0038. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF or DOD.

References

1. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilis-tic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge (1998)
2. Baroni, M., Matiasek, J., Trost, H.: Unsupervised Discovery of Morphologically Related Words Based on Orthographic and Semantic Similarity. In: ACL Special Interest Group in Computational Phonology in Cooperation with the ACL Special Interest Group in Natural Language Learning (SIGPHON/SIGNLL), Philadelphia, Pennsylvania, pp. 48–57 (2002)
3. Wicentowski, R.: Modelling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework. Ph.D. Thesis, The Johns Hopkins University, Baltimore, Maryland (2002)
4. Feng, D.F., Doolittle, R.F.: Progressive Sequence Alignment as a Prerequisite to Correct Phylogenetic Trees. Journal of Molecular Evolution 25(4), 351–360 (1987)
5. Gotoh, O.: Significant Improvement in Accuracy of Multiple Protein Sequence Alignments by Iterative Refinement as Assessed by Reference to Structural Alignments. Journal of Molecular Biology 264(4), 823–838 (1996)
6. Monson, C., Hollingshead, K., Roark, B.: Simulating Morphological Analyzers with Stochastic Taggers for Confidence Estimation. In: 10th Workshop of the Cross-Language Eval-uation Forum, CLEF 2009, Corfu, Greece, Revised Selected Papers. LNCS, Springer, Heidelberg (2010)
7. Kurimo, M., Virpioja, S., Turunen, V.T., Blackwood, G.W., Byrne, W.: Overview and Results of Morpho Challenge 2009. In: 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece, Revised Selected Papers. LNCS, Springer, Heidelberg (2010)
8. Creutz, M.: Induction of the Morphology of Natural Language: Unsupervised Morpheme Segmentation with Application to Automatic Speech Recognition. Ph.D. Thesis, Computer and Information Science, Report D13, Helsinki, University of Technology, Espoo, Finland (2006)
9. Trón, V., Gyepesi, G., Halácsy, P., Kornai, A., Németh, L., Varga, D.: Hunmorph: Open Source Word Analysis. In: ACL Workshop on Software (2005)
10. Varga, D., Halácsy, P., Kornai, A., Németh, L., Trón, V., Váradi, T., Sass, B., Bottyán, G., Héja, E., Gyarmati, Á., Mészáros, Á., Labundy, D.: Hunglish Corpus (2009), `http://mokk.bme.hu/resources/hunglishcorpus` (accessed on August 18, 2009)

Author Index