The Workshop Programme

Malta, Saturday 22nd May 2010

9:00-9:15	**Richard Sutcliffe and Udo Kruschwitz** *Web Logs and Question Answering: Welcome and Overview*
9:15-9:40	**Raffaella Bernardi and Manuel Kirschner** *From artificial questions to real user interaction logs: Real challenges for Interactive Question Answering systems.*
9:40-10:05	**Johannes Leveling** *A Comparative Analysis: QA Evaluation Questions versus Real-world Queries*
10:05-10:30	**Zhicheng Zheng, Yang Tang, Chong Long, Fan Bu and Xiaoyan Zhu** *Question Answering System Based on Community QA*
10:30-11:00	*Coffee Break*
11:00-11:15	**Julia Maria Schulz** *Log-Based Evaluation Resources for Question Answering*
11:15-11:40	**Saeedeh Momtazi and Dietrich Klakow** *Yahoo! Answers for Sentence Retrieval in Question Answering*
11:40-12:05	**Sharon G. Small and Tomek Strzalkowski** *(Tacitly) Collaborative Question Answering Utilizing Web Trails*
12:05-12:30	**Richard Sutcliffe, Kieran White and Udo Kruschwitz** *Named Entity Recognition in an Intranet Query Log*
12:30-13:00	*Discussion*

Workshop Organisers

Richard Sutcliffe
Dept. Computer Science and Information Systems, University of Limerick
Limerick, Ireland
Richard.Sutcliffe at ul.ie

Udo Kruschwitz
School of Computer Science and Electronic Engineering, University of Essex
Wivenhoe Park, Colchester, CO4 3SQ, UK
udo at essex.ac.uk

Thomas Mandl
Dept. Information Science and Language Technology, University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany
mandl at uni-hildesheim.de

Workshop Programme Committee

- Bettina Berendt, Katholieke Universiteit Leuven, Belgium
- Gosse Bouma, Rijksuniversiteit Groningen, The Netherlands
- Paul Clough, University of Sheffield, UK
- Giorgio Di Nunzio, University of Padoa, Italy
- Jim Jansen, Pennsylvania State University, USA
- Udo Kruschwitz, University of Essex, UK
- Johannes Leveling, Dublin City University, Ireland
- Thomas Mandl, University of Hildesheim, Germany
- Fabrizio Silvestri, ISTI-CNR, Italy
- Tomek Strzalkowski, SUNY Albany, USA
- Richard Sutcliffe, University of Limerick
- José Luis Vicedo, University of Alicante, Spain
- Kieran White, University of Limerick, Ireland

Workshop Homepage

http://www.csis.ul.ie/wlqa2010/

Table of Contents

Author Index

Web Logs and Question Answering

Richard F. E. Sutcliffe, Udo Kruschwitz, Thomas Mandl

Dept. Computer Science and Information Systems, University of Limerick
Limerick, Ireland

School of Computer Science and Electronic Engineering, University of Essex
Wivenhoe Park, Colchester, CO4 3SQ, UK

Dept. Information Science and Language Technology, University of Hildesheim
Marienburger Platz 22, 31141 Hildesheim, Germany

E-mail: Richard.Sutcliffe at ul.ie , udo at essex.ac.uk, mandl at uni-hildesheim.de

Abstract

This article briefly characterises the fields of Question Answering and Web Log Analysis, and summarises the main achievements and research methods of each. It then goes on to discuss links between these fields and to describe possible research questions which could be investigated. Finally, it summarises the content of the accepted papers at the workshop and relates these papers to the research questions.

1. Introduction

A Question Answering (QA) system takes as input a short natural language query and gives back an exact answer to it, extracted from a document collection (Hirschman and Gaizauskas, 2001). The origins of QA under this definition can be traced back to TREC-8 (Voorhees and Harman, 1999) when the first QA task was organised. Since then, there have been numerous conferences and workshops concerned with this important field.

A Web Log is a record of a person's internet search; invariably it specifies what query they typed in, and it may include many other pieces of related information such as what sites they subsequently visited, how they subsequently modified their query and so on. Web Log Analysis (WLA) is the field which seeks to examine logs by various means and to use the information gained to improve search engines. One of the earliest studies in this area was Jansen, Spink and Saracevic (2000).

What, then, have these fields in common and what can they learn from each other? Will the fields converge, and what will be the next issues to investigate? The purpose of the Web Logs and Question Answering (WLQA) workshop is to answer these and related questions. In this introductory article we adopt the following strategy. Firstly we give a brief overview of QA and WLA, outlining their history and the key techniques used. Secondly, we state briefly what we consider to be the key areas which are common to both, the principal research questions in these areas, and the means by which they might be investigated. In the final section we summarise the contents of the workshop papers and attempt to fit them into the above picture.

2. Question Answering

2.1 Aims and Background

As understood in the modern sense, a QA system takes as input a short natural language question and produces as output an exact answer, extracted from a document collection. The first time systems of this kind were developed or discussed was probably at the Text REtrieval Conference (TREC) where a QA track was introduced at TREC-8 (Voorhees and Harman, 1999). Up until that time, TREC was mostly concerned with the development and evaluation of Information Retrieval (IR) systems using a common framework for the evaluation and a shared document collection. The QA track inherited these ideas, the key difference being that in QA an exact answer was expected while in IR a list of documents was produced. In order to achieve this exactness, a vital idea was that of the Named Entity (NE) - a piece of information such as a person name, company name, time, date, place etc which could be recognised in a text and hence become a candidate exact answer. The NE idea had been recently developed and used at the Message Understanding Conference evaluations in Information Extraction beginning in 1987 (Grishman, 1996).

In passing we should mention that QA under other definitions goes back to the earliest days of artificial intelligence research. For example, Simmons (1965) discusses fifteen different systems. A full history can be found in Hirschman and Gaizauskas (2001).

Since TREC, QA research has become a very active area and as a result of this work, highly sophisticated systems such as TrueKnowledge (2010) and WolframAlpha (2010) have started to appear.

2.2 Key Techniques in QA

While different approaches to QA have been taken, many systems have converged on a standard architecture. Prager (2006) gives three components: Question Analysis,

Search and Answer Extraction. Other authors give more components by subdividing these further, but the essence is the same.
During Question Analysis, the type of question and the form of NE expected as the answer are determined. For example if we have 'Who is the president of the USA?' then the type of the question is WHO and the expected answer type (i.e. NE) is PERSON. During the Search stage, the document collection is scanned for documents which might contain the answer to the question. The simplest approach to this task involves creating and submitting an IR query based on keywords derived from the question. Finally, in Answer Extraction, candidate NEs of type PERSON are selected from likely documents, and the best one is returned as the answer.
Concisely stated, a number of techniques have proved valuable and replicable in pursuit of the above stages. Firstly, NE recognition is a key process because it must be highly accurate and avoid confusions, for example between person names and company names which are often identical. Initially, approaches revolved around the use of lists (e.g. country names etc) or grammatical analysis (e.g. for dates and times which have an internal structure). In recent years, however, attention has shifted to machine learning approaches which generalise their knowledge from a set of training examples.
Secondly, Predictive Annotation (Prager et al., 2000) allows documents to be retrieved in the Search phase which are guaranteed among other things to contain an NE of the required type. This is done by including pseudo-words in the source documents such as $NAME which are then indexed by the search engine and can be used at query time.
Early Answering (Clarke, 2002) was a reaction to certain TREC questions which asked about Baseball scores, State Flowers and other standard pieces of information which could be determined in advance without looking at the documents at all. The solution was to organise such information in tables and then find the answer directly.
Another key idea, was the logical connection within a text of key elements, rather than their simple co-occurrence (Moldovan et al., 2008). Inference could be based on knowledge structures e.g. derived from WordNet (Fellbaum, 1998) or achieved by word chaining. Related to this was Answer Validation - checking an answer following extraction to see which it was likely to be correct. An influencial approach was that of Magnini et al. (2002) which exploited the redundancy of the web.
Another approach to the use of redundancy in QA was to combine multiple 'pipeline' architectures of the type outlined above into one system (Ahn et al., 2005). This allowed several different mechanisms to be used for producing answer candidates, and several to to be used for scoring them. For an extensive review of QA, see Prager (2006).
Finally, it should be mentioned that a 'second generation' approach to QA involves the representation of documents in a structured and analysed form, achieved during indexing rather than at query processing time. Examples of such systems are START (2010) from MIT, and Powerset (2010).

2.3 Strengths and Weaknesses of QA

In the above we have attempted to summarise activity in a large field over a number of years. What then are the key achievements and shortcomings of all this work? First of all, the tracks at TREC and their closely related counterparts at CLEF (Peñas et al. 2010) and NTCIR (Mitamura et al. 2008) have resulted in a wide understanding of how to build efficient and effective QA systems of the type needed to answer simple factoid questions and others of closely related types. Moreover, work at CLEF and NTCIR has shown that these ideas can be transferred over very effectively to monolingual QA systems in languages other than English.
However, there are also significant weaknesses. The range of questions has been extremely narrow and these have been asked against fixed document collections because this is convenient and practical rather than realistic. In addition, the evaluation paradigm incorporating the judgements Right-Wrong-ineXact-Unsupported has been widely followed for the same reasons. In consequence, QA has in the main tended to ignore the questions of real users or the importance of dialogue as a fundamental basis for answering real questions. Instead, questions have been back-enabled from convenient sources such as newspapers or the Wikipedia.
Due to significant performance improvements at TREC over the years, QA has come to be regarded as a solved problem where no more research can be usefully conducted. However, monolingual factoid QA is only a small part of the overall question answering problem whose solution is essential to the aim of making machines more usable. (Cross-lingual QA has also shown to be a completely unsolved problem.) What is needed therefore are new ideas and new directions. This is the key rationale for the current workshop.

3. Query Log Analysis

3.1 Aims and Background

A query log is a record of queries entered into an internet search engine, together in some cases with additional information such as sites visited etc. According to Clough (2009) quoting Mat-Hassan and Levene (2005), the objectives of Query Log Analysis (QLA) are

- To investigate a searcher's performance;
- To establish the profile of an effective searcher;
- To establish a user's searching characteristics;
- To understand a user's navigational behaviour.

A key starting point for QLA was a panel entitled 'Real life information retrieval: a study of user queries on the Web' at SIGIR in 1997. A landmark paper appeared as a follow-up: Jansen, Spink and Saracevic (2000). The focus of this paper was a log of the Excite engine containing 51,473 entries. The authors conducted a manual analysis of sessions, queries, terms within queries, characteristics of the user and an analysis of failed queries to identify trends within user mistakes. Since that paper there has been a growing interest in QLA.
LogCLEF was first evaluation campaign track focusing on logs. It started at CLEF 2009 (Mandl et al., 2010). The goals were to understand search behaviour especially in multilingual contexts and, ultimately, to improve search

systems. There were two tasks, geographic query identification and library search use. For the former task, logs were obtained from the Tumba! search engine and from the European Library search engine. The purpose was to identify geographical entities in the queries. For the latter task, just the library log was used. Each group carried out a different investigation, including finding translations of queries, searching for documents via search terms previously used to find them, query re-formulation, and analysing characteristics of sequences of queries.

Prior to LogCLEF there was also a log-file based task within GeoCLEF 2007 which used the MSN query log (Mandl et al. 2008).

In 2009, Jansen, Spink and Taksa published a comprehensive handbook on QLA, summarising much of the research which has so far been conducted.

Also in 2009, an important workshop took place entitled Query Log Analysis (Clough, 2009). Some of the main techniques in current use were described there.

Finally, new conferences devoted to QLA and related topics have been established, including WSDM (2008) and WSCD (2009).

3.2 Key Techniques of QLA

Some key approaches to QLA are outlined here. Firstly, the most fundamental form of analysis gathers information such as the numbers of sessions, terms etc and the production of statistics based on these. Jansen et al. (2000) was mainly of this type, and many papers have followed.

Secondly, there have been manual analyses of queries in small numbers looking for detailed aspects such as query type or focus. These sorts of studies have been highly informative but they are limited in the number of queries which can be examined.

Thirdly, there have been automatic analyses which nevertheless do not use Machine Learning (ML) algorithms. For example, Bernstram, Herskovic and Hersch (2009) categorise queries by mapping terms within them onto a domain specific ontology.

Fourthly, ML algorithms have been adopted to carry out tasks such as query classification. Here are several examples of this type of work.

Taksa, Zelikovitz and Spink (2009) show how short queries can be categorised by exploiting information gleaned using IR techniques, a method previously used by Sarawagi (2005).

For topic identification, Ozmutlu, Ozmutlu and Spink (2009) identify three classes of supervised ML algorithm which are effective: Maximum Entropy models, Hidden Markov models and Conditional Random Fields.

Levene (2009) advocates the use of Support Vector Machines (supervised ML) for the classification of queries. He also points out that queries need enriching with result pages or snippets, with related queries and with a training set of categorised web pages. Another line of work has been the prediction of the next link which a user will follow, using Markov chains constructed from logs.

Fifthly, there has been a focus on clickthrough data in the context of web search. Some logs specify what URLs a user clicked when they were returned in response to a query by a search engine. These tell us something important, most obviously that the user was interested enough in a link to click on it. Studies focus on the relation of query and page content, the time spent on result pages and the behaviour on the search engine result page. Radlinksi et al. (2008) is one of many studies concerned with the analysis of such data. Murdock et al. (2009) also use clickthrough data to try to predict which is the best advertisement to show following a query in Yahoo!. They use a Binary Perceptron for this task.

Sixthly, in parallel with the above, there have been a number of studies involving people who participate in a carefully designed experimental study in query formulation etc. In all such cases the logs are captured though the numbers of queries involved tends to be small. For an extensive review of QLA, see Silvestri (2010).

3.3 Strengths and Weaknesses of QLA

The key strength of QLA is that there is potentially a huge amount of data available which is being generated in a naturalistic way, without the users being aware that they are being monitored. This differs greatly from most QA work so far, where queries are generated manually and are therefore not naturally occurring.

The main weakness of QLA is perhaps this same point - the huge amount of data. It cannot be analysed manually and because of its relatively sparse nature (just the query typed, the sites visited etc) we can never know for certain what a user really intended. This can only be inferred, and not with certainty. We do observe user behaviour in log analysis but at much lesser detail than we could observe it in a test environment. Information need, satisfaction and opinion about result documents can only be guessed from logs.

Another difficulty is that search engine companies are reluctant to release their logs for research purposes following the AOL incident in which personal information about people was accidentally put in the public domain (AOL, 2010). However, this can be overcome by projects such as the Lemur Query Log Toolbar which allows users intentionally to have their queries logged (Lemur, 2010).

4. QA & QLA - Common Areas and Research Questions

In this section we try to state briefly what we consider to be the key areas which are common to both, the principal research questions in these areas, and the means by which they might be investigated.

First of all, how real are the questions in QA and the queries in QLA? In QA, the questions are not usually from real users, they are devised by the assessors at CLEF, TREC etc. Secondly, they are restricted to certain well-known simple types which are only a small subset of the real questions which people wish to ask. These simplifications are necessary due to the limitations of our present day QA systems. Thirdly, questions are considered in isolation (or in some tracks a fixed group) and not in a dialogue context whereas in our interactions with people all questions are answered in context and with the possibility for clarification (see however Webb and

Webber, 2009 on interactive QA).
On the QLA side, queries are real and they are numerous. On the other hand, only very few (perhaps 1%) are actual queries (de Rijke, 2005) and for the others we cannot be sure of the true intent.
Second, we list eight key questions in relation to QA and QLA:

1. Can the meaning of IR queries in logs be deduced automatically in order to extract the corresponding questions from them? Can appropriate answers be suggested to users after the retrieval of result documents?

2. Can NLP techniques developed within QA, e.g. Named Entity recognition be applied to the analysis of query logs?

3. Can logs be used to deduce useful new forms of question (i.e. not simple factoids) which could be looked at next by QA researchers?

4. Can questions grouped into sessions be comprehended in such a way as to deduce the underlying implicit natural language dialogue consisting of a coherent sequence of questions where each follows logically from both the previous ones and the system's responses to them?

5. Are there logs from real (or experimental) QA systems like lexxe.com and what can be learned from them from the perspective of designing evaluation tasks? What about logs from sites like answers.com (where queries are answered by human respondents)?

6. Are QA query logs different from IR query logs? Do users behave differently in QA systems?

7. Can click-through data - where the aim of a question can be inferred from the returned documents which are inspected - be used for the development of QA systems for example for the deduction of important query types and their links to IR queries?

8. Are there logs of transcribed speech made from telephone QA systems and what analysis could be carried out on those, using for example techniques developed at related tracks at CLEF such as Cross-Language Speech Retrieval (CL-SR) and Question Answering on Script Transcription (QAST)?

5. Summary of the Workshop Papers

In this final section, we outline the main contributions of the papers accepted for the workshop and we attempt to link them together. These contributions address some of the research questions posed above.

Bernardi and Kirschner - *From artificial questions to real user interaction logs: Real challenges for Interactive Question Answering systems*.

This paper focuses on the issue of real logs vs. not real QA questions at TREC etc. There are three question sets: TREC, Bertomeu (collected in a controlled Wizard-of-Oz study) and BoB (a chatbot working at a university library site). These are analysed in respect of several different measures comparing utterences in a QA dialogue. The main conclusion is that the TREC data differs significantly from BoB in important respects such as length of query (BoB queries are shorter) and number of anaphora (Bob queries have less). The thinking is that in future TREC-style evaluations, questions should take into account these factors to make them as realistic as possible.

Leveling - *A Comparative Analysis: QA Evaluation Questions versus Real-world Queries.*

This paper compares queries submitted to a web search engine, queries submitted to a Q&A service (answers.com), and those used at TREC and CLEF in the QA tracks - six collections in all. This is very interesting because it is a direct comparison between the QA side and the QLA side. The core of the paper deals with an experiment in which well formed questions from answers.com are converted into IR-style queries (e.g. using just content words) and then a naive Bayes classifier is used to try to recover the expected answer type and the original wh-word frame. For example "capital Ethiopia" should become "What is the capital of Ethiopia" and the answer type is capital city. The thinking behind this interesting study is that if log queries can be converted to questions they can be answered exactly by a QA system.

Zhu et al. - *Question Answering System Based on Community QA*.

This paper considers whether sites such as Yahoo Answers - which contain millions of submitted questions and answers to them - can be used as a log-like resource to improve question answering. Given and input query, similar queries are then identified in the logs and their answers retrieved. A summarisation algorithm is then used to select sentences from the retrieved answers which can be used as the response to the input query.

Momtazi and Klakow - *Yahoo! Answers for Sentence Retrieval in Question Answering*.

This paper is also concerned with Yahoo Answers and its potential for improving QA performance. The authors developed two statistical frameworks for capturing relationships between words in a question-answer pair within Yahoo Answers. These were then used in a sentence selection task using as input TREC 2006 queries. Their best results exceeded the baseline which was considered to be the word based unigram model with maximum likelihood estimation.

Small and Strzalkowski - *(Tacitly) Collaborative Question Answering Utilizing Web Trails*.

The aim of the work is to study logs made by monitoring users in an interactive QA study. The information saved includes the question answered, the responses given and those documents actually saved by each participant. Documents saved are placed in a standard order to allow comparisons between different searchers working on the same task. The key result is that in a study of 95 episodes, there is quite a degree of overlap between sets of files saved. This suggests an opportunity for sharing of data. One possible means of doing this is to observe a sequence of documents saved by a user, and when it overlaps with a previously observed sequence produced by another user, to offer the remainder of that saved sequence. This paper is interesting because it is the only one which collects QA data in a naturalistic setting, albeit within a controlled experiment where users are given predetermined tasks.

Sutcliffe, White and Kruschwitz - *Named Entity Recognition in an Intranet Query Log*.

This paper is concerned with queries in a highly focused log of searches conducted at a university web site. The authors firstly conducted a manual study some queries, categorising each by topic. In the process, a list of important named entity types was created. Secondly, training data for each NE type was created from the university website and this was used to train a maximum entropy NE tagger on a much larger log. This was evaluated, and statistics concerning NE occurrences in the log as a whole were computed. Finally, the possible use of NE data in answering the queries is discussed.

Mandl and Schulz - *Log-Based Evaluation Resources for Question Answering*.

This paper is concerned with the relationship between query logs and well-formed questions as answered by QA systems. The authors propose a system which can switch between IR-mode and QA-mode, depending on the input. They first discuss some of the log resources which are available for this kind of work, together with related Log analysis tracks at CLEF, and then present a preliminary analysis of question-like queries in the MSN log.

6. Conclusion

QLA and QA appear to be fields which intersect in a number of ways which suggest new research challenges for both. We listed some of these in Section 4. The papers accepted for the workshop address some of these challenges but not all of them.

The paper by Bernardi and Kirschner compares real and created questions and shows how users really behave in QA systems as opposed to how evaluation designers think they behave (research question 6). A related paper by Leveling also compares the two kinds of questions. In addition, it tries to learn how the information need can be deduced from a short IR query (research question 1). However, a related aim, to use logs to develop better evaluation sets (research question 3) has not been addressed in depth.

The study of Momtazi and Klakow takes a deeper look at a real world QA service and its logs (research question 5). The same goal is pursued by Zhu at al. but they apply a different algorithm. A user study concerning a complex information need is presented by Small and Strzalkowski. Theirs is the only approach which addresses the issue of user sessions (research question 4). Sutcliffe at al. show for a real log file how NLP techniques can support the analysis (research question 2). Finally, research question 3 is again addressed by a short paper of Mandl and Schulz which argues that current logs for IR evaluation can also be a source of QA style questions.

The issues of click-through data and spoken language QA (research questions 7 and 8) have not been addressed at this workshop. Indeed, the other research questions have not generally been addressed in depth. However, several contributions used real-world QA resources and dealt with their properties in general and their differences from QA evaluation resources.

In summary, the papers provide an interesting survey of work in this developing field. However, while much has been achieved, all the contributors suggest interesting and worthwhile avenues for further research linking QLA and QA.

7. References

Ahn, D., Jijkoun, V., Müller, K., de Rijke, M., Schlobach, S., Mishne, G. (2005). Making Stone Soup: Evaluating a Recall-Oriented Multi-stream Question Answering System for Dutch. In: Peters, C., Clough, P., Gonzalo, J., Kluck, M., Jones, G., Magnini, B. (eds): *Multilingual Information Access for Text, Speech and Images: Results of the Fifth CLEF Evaluation Campaign.* Berlin et al.: Springer [Lecture Notes in Computer Science 3491] pp. 423--434.

AOL (2010). http://en.wikipedia.org/wiki/AOL_search_data_scandal. Accessed 2010.

Bernstram, E.V., Herskovic, J. R., Hersch, W. R. (2009) Query Log Analysis in Biomedicine. In J. Jansen, I. Taksa & A. Spink (Eds.) *Handbook of Web Log Analysis* (pp. 329-344). Hershey, PA: IGI Global.

Clarke, C. L. A., Cormack, G. V., Kemkes, G., Laszlo, M., Lynam, T.R., Terra, E. L., Tilker, P. L. (2002). Statistical Selection of Exact Answers (MultiText Experiments for TREC 2002). In *Proceedings of TREC 2002*. http://trec.nist.gov/pubs/trec11/t11_proceedings.html

CLEF (2010). http://www.clef-campaign.org. Accessed 2010.

Clough, P. (2009). Query Log Analysis Workshop 2009 (slides). http://ir.shef.ac.uk/cloughie/qlaw2009/presentations/clough.pdf

Clough, P., Berendt, B. (2009). Report on the TrebleCLEF query log analysis workshop 2009, *ACM SIGIR Forum*, 43(2), pp. 71-77.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database and Some of its Applications*. Cambridge, MA: MIT Press.

Grishman, R., Sundheim, B. (1996). Message Understanding Conference - 6 : A Brief History. www.aclweb.org/anthology/C/C96/C96-1079.pdf

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

Hoekstra, D., Hiemstra, D., van der Vet, P., Huibers, T. (2006). Question Answering for Dutch: Simple does it. *Proceedings of the BNAIC: Benelux Conference on Artificial Intelligence*.

Jansen, B. J., Spink, A., Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the web. *Information Processing & Management*. 36(2), pp. 207-227.

Jansen, J., Taksa, I., Spink, A. (Eds.) (2008). *Handbook of Web Log Analysis*. Hershey, PA: IGI Global.

Lemur (2010). http://www.lemurproject.org/querylogtoolbar/. Accessed 2010.

Magnini, B., Negri, M., Prevete, R., Tanev, H. (2002). Is It the Right Answer? Exploiting Web Redundancy for Answer Validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Mandl, T., Agosti, M., Di Nunzio, G., Yeh, A., Mani, I., Doran, C., Schulz, J. M. (2010). 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.): *Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments: Proceedings 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece. Revised Selected Papers.* Berlin et al.: Springer [Lecture Notes in Computer Science] to appear. Preprint in Working Notes: http://www.clef-campaign.org/2009/working_notes/LogCLEF-2009-Overview-Working-Notes-2009-09-14.pdf

Mandl, T., Gey, F., Di Nunzio, G., Ferro, N., Larson, R., Sanderson, M., Santos, D., Womser-Hacker, C., Xing, X. (2008). GeoCLEF 2007: the CLEF 2007 Cross-Language Geographic Information Retrieval Track Overview. In: Peters, C.; Jijkoun, V.; Mandl, T.; Müller, H.; Oard, D.; Peñas, A.; Petras, V.; Santos, D. (Eds.): *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum. CLEF 2007, Budapest, Hungary, Revised Selected Papers*. Berlin et al.: Springer [Lecture Notes in Computer Science 5152] pp. 745--772.

Mansourian, Y., Madden, A. D. (2007). Methodological approaches in web search research. *The Electronic Library*, 25(1), pp. 90-101.

Mat-Hassan, M., Levene, M. (2005). Associating search and navigation behavior through log analysis. *JASIST*, 56(9), pp. 913-934.

Maybury, M. T. (Ed.) (2004). *New Directions in Question Answering.* Cambridge, MA : MIT Press.

Mitamura, T., Nyberg, E., Shima, H., Kato, T., Mori, T., Lin, C.-Y., Song, R., Lin, C.-J., Sakai, T., Ji D., Kando, N. (2008). *Overview of the NTCIR-7 ACLIA Tasks: Advanced Cross-Lingual Information Access. In: Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access* http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings7/pdf/revise/01-NTCIR7-OV-CCLQA-MitamuraT-revised-20090113.pdf

Moldovan, D., Pasca, M., Surdeanu, M. (2008). In T. Strzalkowski and S. Harabagiu (Eds.) *Advances in Open Domain Question Answering* (pp. 3-34). New York, NY : Springer.

Murdock, V., Ciaramita, M, Plachouras, V., Garcia, L, Olivares, X., van Zwol, R. (2009). Online learning from Click Data (slides). http://ir.shef.ac.uk/cloughie/qlaw2009/presentations/murdock.pdf

NTCIR (2010). http://research.nii.ac.jp/ntcir/. Accessed 2010.

Peñas, P., Forner, P., Sutcliffe, R., Rodrigo, A., Forăscu, C., Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P. (2010): Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. In: Peters, C., Di Nunzio, G., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (Eds.): *Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments: Proceedings 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece. Revised Selected Papers.* Berlin et al.: Springer [Lecture Notes in Computer Science] to appear. Preprint in Working Notes: http://www.clef-campaign.org/2009/working_notes/ResPubliQA-overview.pdf

Powerset (2010). http://www.powerset.com/. Accessed 2010.

Prager, J. (2006). Open Domain Question-Answering. *Foundations and Trends in Information Retrieval* 1(2), pp. 91-231.

Prager, J., Brown, E. W., Coden, A., Radev, R. (2000). Question Answering by Predictive Annotation. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR00, Athens, Greece, pp. 184-191.

QLA Workshop (2009). http://ir.shef.ac.uk/cloughie/qlaw2009.

Radlinski, F. Kurup, M., Joachims, T. (2008). How Does Clickthrough Data Reflect Retrieval Quality? In Proceedings of the ACM Conference on Information and Knowledge Management (CIKM).

de Rijke, M. (2005). *Question Answering: Now and Next*. Invited Talk, University of Twente, October 2005.

Silvestri, F. (2010). Mining Query Logs: Turning Search Usage Data into Knowledge, Foundations and Trends in Information Retrieval, 4(1-2), 1-174.

Simmons, R. F. (1965). Answering English Questions by Computer : A Survey. *Communications of the ACM*, 8(1), pp. 58-65.

START (2010). http://start.csail.mit.edu/ Accessed 2010.

Strzalkowski, T., Harabagiu, S. (Eds.) (2008). *Advances in Open Domain Question Answering*. Dordrecht, The Netherlands: Springer.

TREC (2010). http://trec.nist.gov/. Accessed 2010.

TrueKnowledge (2010). http://www.trueknowledge.com/. Accessed 2010.

Voorhees, E., Harman, D. (1999). Overview of the Eighth Text REtrieval Conference. In *Proceedings of the Eighth Text REtreival Conference, Gaithersburg, Mayland, November 16-19*, pp. 1-33.

Webb, N., B. Webber (Eds.) (2009). *Journal of Natural Language Engineering* (Special Issue on Interactive

Question Answering), 15(1), pp. 1-141.
WolframAlpha (2010). http://www.wolframalpha.com/. Accessed 2010.
WSCD (2009). http://research.microsoft.com/en-us/um/people/nickcr/wscd09/.
WSDM (2008). http://wsdm2009.org/wsdm2008.org/index.html

From artificial questions to real user interaction logs: Real challenges for Interactive Question Answering systems

Raffaella Bernardi and **Manuel Kirschner**

KRDB, Faculty of Computer Science
Free University of Bozen-Bolzano, Italy

Abstract

Much research in Interactive Question Answering (IQA) has centered on artificially collected series of context questions. Instead, the goal of this paper is to emphasize the importance of evaluating IQA systems against *realistic* user questions. We do this by comparing the highly popular TREC QA context task data against two more realistic data sets: firstly, a corpus of real user interaction logs that we collected through a publicly accessible chat-bot, and secondly, a corpus of QA dialogues collected in a Wizard-of-Oz study. We compare these data using basic quantitative measures and different measures for expressing inter-utterance coherence. We conclude with proposals for choosing test data for a new evaluation campaign that is centered on realistic user-system interactions, and that is well suited for empirical and Machine Learning approaches.

1. Introduction

Question Answering (QA) systems have reached a high level of performance within the *single, factoid question* scenario originally defined by the TREC QA competitions. As a consequence, the research community has moved on to tackle new challenges, as shown by the *context question* and *Interactive QA (IQA)* tasks proposed in recent instantiations (Voorhees, 2004). The idea of extending "single shot" questions to context QA first made its way into the 2001 QA track (Voorhees, 2001), in the so-called context task, which consisted of 10 question series of around 5 topically related questions each. From TREC 2004 (Voorhees, 2004) onward, the *main* QA task was changed to consist of series of questions, where for each series a so-called *target* string explicitly identified the topic that is common to all the questions from a particular series. Examples of TREC'01 and TREC'04 series of questions are shown below.

1. In what country was the first telescope to use adaptive optics used?
2. Where in the country is it located?
3. What is the name of the large observatory located there?

Table 1: Sample from TREC'01

Series question target: Hale Bopp comet
1. When was the comet discovered?
2. How often does it approach the earth?
3. In what countries was the comet visible on its last return?

Table 2: Sample from TREC'04

The yearly TREC QA track – run by NIST from 1999 through 2007 (e.g., (Dang et al., 2007)) – has helped the QA community share results, and lead to new techniques being embraced much faster.

In a report known as the ARDA QA Roadmap (Burger et al., 2000), a number of researchers from the QA community suggested several important research challenges for QA. Two of the challenges mentioned were Context QA IQA, both of which were later addressed in specific tracks of the TREC QA task, and thus have had a major influence on current QA research in general.

For the case of Context QA, (Burger et al., 2000) see the role of context as that of clarifying a Follow-Up Question (FU Q), resolving ambiguities, or keeping track of an investigation performed through a series of questions. The underlying motivation is that in a real information-seeking scenario, questions are not asked in isolation; instead, users ask FU Qs that might relate in different ways to the ongoing dialogue. In this work, we confirm this claim empirically by analyzing the corpus of real user interaction logs that we have collected through BoB, a chat-bot that has been providing library help-desk information on our University Library web-site[1] for over one year.

As for IQA, (Burger et al., 2000) foresee that a questioner might want not only to reformulate a question, but to engage in a real user-system dialogue; thus, IQA tries to go a step beyond Context QA, towards a truly intelligent system for humans trying to access information in an efficient way. In practice, the term *IQA* has been used to denote quite different extents of dialogue capabilities of a QA system. One example is the TREC complex Interactive QA (cIQA) task, conducted as part of the 2006 TREC QA track (Kelly and Lin, 2007): in an approach inherited from interactive IR, assessors had to provide one iteration of user-system feedback in the form of relevance feedback; the system was then supposed to take this feedback into account and provide a new and improved answer to the user. Besides cIQA, there has been considerable work on IQA systems with even more extensive interactive capabilities. Such systems were characterized in the following ways by contributors of a recent, influential IQA Workshop (Interactive Question Answering Workshop at HLT-NAACL, 2006): the system draws the user into a conversation (Strzalkowski, 2006); the

[1] `http://www.unibz.it/en/library`

system understands what the user is looking for, what the user has done and what the user knows (Kelly et al., 2006); the system is a partner in research. Other recent approaches considering highly interactive IQA systems can be found in (Maybury, 2003), (Strzalkowski and Harabagiu, 2006) and (Mitkov et al., 2009).

In this work, we propose a definition of *IQA* that includes *Context QA* (as in the TREC QA context track), but puts emphasis on the availability of *realistic* user questions, and on the existence of system answers in the dialogue context. We believe it is crucial that a large enough amount of IQA dialogue data are easily available for empirical and Machine Learning-based research in IQA: this criterion was met in the case of the TREC Context QA data, but not in the case of TREC cIQA data, nor of much of the above-mentioned literature proposing more sophisticated, highly interactive IQA systems. The BoB dialogue data described in this paper try to strike a balance between the two goals: the availability of IQA data for empirical research, combined with an adequate level of realism and naturalness of the IQA dialogues.

The goal of this paper is to emphasize the importance of evaluating (Interactive) QA systems against *realistic* user questions. In order to highlight aspects of real user interactions with an IQA system, we compare the TREC context task data against two data sets: firstly, a corpus of QA dialogues collected semi-artificially in a Wizard-of-Oz study (Bertomeu, 2008), and secondly, BoB's real user interaction logs (Kirschner, 2010).

We compare the data sets by considering basic quantitative measures like dialogue and utterance length, prevalence of anaphoric references, and different measures for quantifying inter-utterance coherence. As for the latter, we consider those measures that we have used successfully as features for modeling IQA dialogue structure, and that improved an IQA system's accuracy in answering FU Qs (Kirschner et al., 2009; Bernardi et al., 2010). In particular, we use a *shallow* feature that quantifies inter-utterance coherence through simple string similarity, and a variety of *deep* features that define coherence based on two existing theories of coherence in Dialogue and Discourse. The first theory (Sun and Chai, 2007), strongly related to the well-known Centering Theory (Grosz et al., 1995), looks at *entity*-based coherence, while the second theory (Chai and Jin, 2004) considers *action*-based coherence, where the verbs of successive user questions are considered.

We now move to describe the different IQA data sets (Section 2.), introduce the inter-utterance coherence features (Section 3.), and the data sets in terms of these features (Section 4.). From this analysis, in Section 5., we draw conclusions that could be useful for setting up a new evaluation campaign for Interactive Question Answering systems that considers *realistic* user questions.

2. Data sets

We now introduce each of the data sets, explaining how it was collected, and providing a qualitative description on the linguistic and stylistic levels. We also provide an example excerpt for each data set. Basic quantitative features of the data sets will be given in Section 4.1..

TREC The TREC data come from the Text REtrieval Conferences question answering (QA) track (Voorhees, 2004), namely from its context task. This task was designed to study contextual, interactive QA by allowing for series of contextually related questions. We use two English language data sets from the 2001 and 2004 editions of the TREC QA track.

As the two data samples in Tables 1 and 2 show, all questions are grammatical sentences and contain no typos, in stark contrast to any real user scenario. Moreover, again differently from a real setting, Follow-up Questions were asked by the NIST staff without knowing the answer to the previous question. Finally, within a series there are no topic shifts, and the FU Qs are always about the same topic (question target) defined for the entire series. All these aspects show how TREC data are rather different from real user-system interactions.

Bertomeu The corpus described in (Bertomeu, 2008) was collected via a Wizard-of-Oz study, involving 33 subjects. Interactions are about language technology, and the information comes from a language technology information database. The subjects were given the following tasks. Through their interactions, they had to find (i) three traineeships at three different projects in three different institutions, (ii) three conferences in the winter term and three conferences in the summer term taking place at different times, on different topics, etc., (iii) information for writing a report on language technology in Europe in the last ten years. Subjects were asked to complete these tasks in one hour. The wizard answered mostly by listing entities from a database, or by asking a clarification question and making statements about the database contents. Table 3 shows an excerpt from a user-wizard dialogue.

Q1:	Can you show me those which are about dialog processing or artificial intelligence?
A1:	The server is busy. Your question will be processed in a moment . . . International Joint Conference on Natural Language Processing.
Q2:	Are all these conferences taking place between now and September 2006?
A2:	yes
Q3:	When does "Translating and the computer 27" take place?
A3:	From 2005-11-24 until 2005-11-25.
Q4:	I'm interested in "International Conference on Image Processing" and "31st International Conference on Acoustics, Speech, and Signal Processing" and "13th Annual ACM International Conference on Multimedia". When do these take place?
A4:	From 2005-11-11 until 2005-11-15; From 2006-05-14 until 2006-05-19; From 2005-11-06 until 2005-11-12.
Q5:	When is the "Second Australian Conference on Artificial Life"?
A5:	From 2005-12-05 until 2005-12-08.

Table 3: Sample from Bertomeu

Differently from TREC data, the interactions with the wizard contain ungrammatical sentences as well as sentences with typos; moreover, the users asked FU Qs after having received an answer. These features make the Bertomeu data more related to real user-system interactions. However, since the subjects were given precise tasks and a time limit to achieve their information goals, the data is inherently not natural.

BoB This data set was collected via the Bolzano Bot (BoB), which has been working as an on-line virtual help desk for the users of our University Library since October 2008.[2] Bob speaks English, Italian and German; in this work, we consider only the English logs. Users ask questions and the system answers them – apart from a few subdialogues, tagged as such, in which BoB asks clarification questions or suggests other relevant information the user might be interested in.

Q1:	how can i request a book that is not in the library
A1:	If we don't have the book you should in any case check the catalogue of the state libraries Claudia Augusta or Tessmann. If it isn't there as well you can order it via our Interlibrary Loan service.
Q2:	explain the interlibrary loan service
A2:	If you need a book or an article which is not available in any library in South Tyrol you can use our Interlibrary Loan Service. Would you like to know more about it?
Q3:	yes
A3:	What exactly do you need? Some general information about the Interlibrary Loan Service? Or do you want to know how to order a book? Or do you need an article?
Q4:	how to order a book
A4:	To order a book we need the following data: the author of the book, the title of the book, and the publication date.

Table 4: Sample from BoB

Like Bertomeu's logs, the interactions with BoB contain many typos and ungrammatical sentences, as well as just keywords. Moreover, the dialogues are rather short in length: while some users seem to use the system to explore library-related information and let themselves be "guided" by BoB, many users seem to have just one information need, and leave the conversation after they asked the relevant question, and hopefully received the correct answer. Moreover, there are several FU Qs that are paraphrases of previous questions. In this case, the user might be trying to refine her question, because the answer was correct but not what the user wanted to know, or the answer was incorrect and the user thinks the system has not understood her question. Another possible cause for rephrased questions is that the user explores the topic further by moving the focus of attention to a new related entity or a new related action, as in the following example: Q_1: *Could you recommend me some book?* Q_2: *Could you recommend me some novel?*. These kinds of interactions seem typical of real user data, and they are also reported in the literature (Bertomeu, 2008; Yang et al., 2006). Like in TREC data, FU Qs that are *Topic Continuations* – i.e, that do not switch to some unrelated new topic – may contain ellipses and anaphora, as in: Q_1: *Where can I find design books?* Q_2: *and dvd?*. We will address this aspect in Section 4..

3. Inter-utterance features

It has been shown that an IQA system generally needs to consider just the immediately preceding interactions (i.e., the previous question and its answer) to answer FU Qs (Kirschner et al., 2009). We converted all IQA dialogues of the data sets described above into what we call *dialogue snippets*, each consisting of four successive utterances: Q_1, A_1, Q_2, A_2. Each snippet thus represents a FU Q, termed Q_2, preceded by the previous user question and system answer, and followed by its (correct) answer A_2.[3] We use this snippet representation to calculate the two types of inter-utterance features described in the following.

3.1. Shallow string similarity feature

We use a shallow feature to measure string similarity between two utterances within a snippet. The idea is that string similarity is a simple approach to measuring *coherence* between two utterances; we want to compare coherence between Q_2 and the preceding utterances across the different IQA dialogue sets to get a first, shallow approximation of how FU Qs relate to the dialogue context.
Our string similarity metric is based on inverse *tf.idf*-distance of the bag of words representations of the two utterances. If two utterances (e.g., Q_1 and Q_2) share some terms, they are similar; the more *discriminative* the terms they share, the more similar the utterances. See (Kirschner, 2010) for a detailed technical description of our implementation of this feature.

3.2. Dialogue and Discourse features

Like the shallow feature introduced above, the Dialogue and Discourse features described in this section measure different types of coherence between utterance within a dialogue snippet. However, now the notion of coherence is based on different theories from the field of Dialogue and Discourse modeling. Following (Sun and Chai, 2007), we consider features describing coherence in terms of repeated occurrences of discourse entities: entity-based coherence. Moreover, following (Chai and Jin, 2004), we define features that describe different *Informational Transitions* holding between a user's previous question and their FU Q, based on the actions (i.e., the verbs) underlying these questions.

3.2.1. Entity-based dialogue coherence

We introduce three features for describing coherence relations between specific pairs of utterances, based on the reference, forward and transition models of (Sun and Chai, 2007). These relations define dialogue coherence by checking for the repetition of certain discourse entities, i.e., noun phrases, within a dialogue snippet. The three relations are inspired by Centering Theory (Brennan et al., 1987; Grosz

[2]We developed the chat-bot web application as an open source project, which we would like to share with the research community interested in collecting similar IQA dialogues. See `http://code.google.com/p/chatterbot-bob`.

[3]Because some of the features described below need to consider also the preceding context of Q_1, we keep information about the order in which the snippets represent the original dialogue.

et al., 1995); more specifically, their definitions build on the following definitions from (Brennan et al., 1987):

Forward-looking centers: each utterance is associated with a list of *forward-looking centers*, consisting of those discourse entities that are mentioned in the utterance.

Preferred center: the list of forward-looking centers is ordered by likelihood of each entity to be the primary focus of the subsequent utterance; the first entity on this list is the *preferred center*.

Our implementation of the three Centering-Theory-based features relies on the automatic detection of forward-looking and preferred centers, and on automatic anaphora resolution. For these tasks, we make use of GuiTAR (Poesio and Kabadjov, 2004; Kabadjov, 2007). Firstly, GuiTAR yields a list of resolved antecedents referred to in a given utterance by anaphora.[4] Secondly, it finds a list of an utterance's forward-looking centers, i.e., any noun phrase directly mentioned in the utterance. In this work, and following (Ratkovic, 2009), we consider the *preferred center* to be that entity from the list of forward-looking centers which is *mentioned first* in the utterance, and which is not a first or second person pronoun.

We use the following approach, proposed in (Ratkovic, 2009), to identify the *preferred center* of each question. For all anaphora found in the question, we use GuiTAR to extract their antecedents, again using the previous questions as context; the first (in terms of linear order) antecedent which is not a first or second person pronoun[5] becomes the *preferred center* of the question. If no preferred center was found so far, the first noun phrase (which is not a first or second person pronoun) appearing in the question itself becomes the preferred center.

Our first feature, `center.Reference`, implements the idea behind the *reference model* of (Sun and Chai, 2007). It is a binary feature that indicates whether a specific coherence relation holds between Q_2 and A_2. First of all, we resolve any anaphora present in Q_2, providing Q_1 as dialogue context. Note that the dialogue context does not include the preceding answers. Although we show in Section 4.2. that these answers are likely locations of antecedents to anaphora found in FU Qs, we do not consider answers in the feature definition for purely practical reasons (discussed in Section 4.2.), to keep our data sets comparable. The `center.Reference` feature evaluates to *true* if the noun phrase head of any antecedent is mentioned in A_2. Note that in our implementation we do not consider cases that are string-identical, thus disregarding all classes of anaphora detected by GuiTAR, but personal pronouns.

The `center.Forward` feature implements the *forward model* of (Sun and Chai, 2007). It is again a binary feature, this time indicating the presence of a specific coherence relation holding between Q_1 and A_2. After resolving anaphora in Q_1 – using Q_2 from the previous dialogue snippet as context – the `center.Forward` feature becomes *true* if either the noun phrase head of any antecedent is mentioned in A_2, or any *forward-looking center* from Q_1 can be found also in A_2.

Finally, the `center.Transition` feature is based on the *transition model* of (Sun and Chai, 2007). It builds on the four discourse transitions between adjacent utterances that Centering Theory introduced (Brennan et al., 1987). Somewhat differently from that classic theory, (Sun and Chai, 2007) define the transitions depending on whether the head and/or the modifier of the *preferred centers* are continued or switched between Q_1 and Q_2.[6] The four possible values of the `center.Transition` feature are defined as follows, based on the two preferred centers of Q_1 and Q_2: *Continue*: both the head and the modifier stay the same. *Retain*: the head stays the same, but the modifier is different. *Smooth shift*: the head is different, but the modifier stays the same. *Rough shift*: both the head and modifier are different.

3.2.2. Action-based dialogue coherence

We use three different features to describe the Informational Transitions proposed by (Chai and Jin, 2004). All these are based on certain relations between the predicate-argument structures of two consecutive user questions, Q_1 and Q_2.

(a) `ConstraintRefinement`: a question concerns a similar topic as that of a previous question, but with different or additional constraints

(b) `ParticipantShift`: the FU Q is about a similar topic but with different participants

(c) `TopicExploration`: the two questions are concerning the same topic, but with different focus

We implemented these features based on the grammatical relations produced by the Stanford parser (Klein and Manning, 2003) in dependency mode (de Marneffe et al., 2006). The main ideas behind the feature implementations are the following (see (Ratkovic, 2009; Bernardi et al., 2010) for more details):

(a) the two questions contain the same syntactic predicate and the same subject or object, but Q_2 has either an *additional* or a *missing* argument (subject, object, adverb, preposition, or adjectival modifier) when compared to Q_1

(b) the two questions have the same syntactic predicate, but either the subject, object or argument of some preposition are different

(c) the two questions have either the same syntactic predicate, subject, object or preposition.[7]

[4] Anaphora considered by GuiTAR are: definite noun phrases, proper nouns, proper nouns with definite articles, and personal pronouns.

[5] Very often in IQA dialogue data the subject of the question is a personal pronoun like "I". This pronoun carries no useful information regarding the informational content of the question, and we thus exclude such pronouns from our algorithm.

[6] Centers are noun phrases. The syntactic structure of a noun phrase comprises a *head noun*, and possibly a *modifier*, e.g., an adjective.

[7] We found this rather lax definition to work best in FU Q classification experiments.

4. Data comparison

4.1. Basic quantitative measures

Differences among the dialogue data sets are already evident by looking at basic statistics underlying the data.
Table 5 provides important quantitative measures. Most evidently, due to the Wizard-of-Oz design, the dialogues in the Bertomeu data are significantly longer than the naturally occurring IQA dialogues of genuinely interested users in the BoB data. Also, the questions are twice as long on average, indicating that users tend to form simpler and shorter queries in an actual IQA system. For the BoB data, note that before extracting dialogue snippets from the 1,161 dialogues containing at least one FU Q, we removed those dialogues where Q_2 is not a library-related question, i.e., where the user did not seem to have an information need.
Looking more in detail at the dialogue lengths in BoB, Table 6 gives the counts and proportions of dialogues with the typical numbers of user questions. In this realistic IQA setting, two thirds of users asked at least one FU Q. The mean number of user questions across all dialogues containing at least two user questions is 5.3.

4.2. Anaphora

To assess the relevance of the dialogue context preceding Q_2, we again resort to GuiTAR for detecting and resolving anaphora; we now compare the resulting anaphora counts across the different IQA data sets. Table 7 lists counts and corresponding proportions out of the total number of Q_2s of each data set. From this table, we note the following: both the TREC and Bertomeu data sets contain proportionally more total anaphora than the realistic IQA data from BoB. The difference in anaphora proportions between BoB and Bertomeu is mostly due to personal pronouns (perspro) and proper nouns (pn): in both categories, Bertomeu contains twice as many anaphora than BoB. On the other hand, both TREC data sets contain a clearly exaggerated proportion of personal pronouns, with respect to both BoB and even Bertomeu. This shows again that the TREC question series can not be take to represent realistic IQA questions.
Although we show in Table 8 that previous answers are likely locations of antecedents to anaphora in questions, we do not provide GuiTAR with the previous answer A_1 as context for purely practical reasons. Firstly, for TREC, answers are not available, and secondly, for Bertomeu, the syntactic parser used by GuiTAR fails on the majority of system answers from that data set, due to their excessive sentence length. However, for the BoB data set, Table 8 does explore the issue of considering A_1 as additional context in the anaphora detection and resolution phase. From this table it is evident that the previous answer plays an important role as a location for antecedents: if GuiTAR considers also A_1 as a potential location of antecedents for anaphora from Q_2, there is a relative increase of 61% of detected anaphora.
Finally, to get a rough estimate of the accuracy of our automatic anaphora detection procedure based on GuiTAR, Table 9 compares automatically detected anaphora against a gold standard hand annotation. We use the two TREC data sets for this purpose. From this table it seems evident that GuiTAR has a problem with recall, i.e., it seems to miss anaphora that were found by the human annotator. We still believe that our automatic procedure serves its purpose as a means for automatically comparing the IQA dialogue sets we are interested in.

4.3. Inter-utterance features

String similarity feature We calculated the shallow, string similarity-based feature as described in Section 3. to express the degree of term-based similarity between two consecutive questions across the different data sets. TREC data contain only topic continuation (TC) FU Qs, whereas BoB logs contain topic shifts (TS) too. Hence, we took a sample of BoB's logs (417 snippets out of 1,522) and marked manually whether the FU Q was a TC or a TS; the sample snippets contain 250 TC and 167 TS FU Qs. In Table 10 we summarize the average of the similarity between a FU Q and its previous question. In the case of BoB, we report string similarity figures of both the whole dialogue corpus, as well as the subset containing only those 250 FU Qs marked as TC.
We make two observations based on this table. Firstly, across all data sets, the transitions between Q_1 and Q_2 have the highest average string similarity of all utterance-utterance combinations. This is a first indication that consecutive questions in IQA often concern similar topics, by way of containing similar terms. As we see from the higher similarity scores of $Q_1.Q_2$ for the TC subset of the BoB data compared to the full BoB data, topic continuation seems to be detectable to some extent already with this simple shallow feature of string similarity. The second observation we draw from this table is that the average string similarity between Q_2 and its correct answer (A_2) is lower for the Bertomeu data set when compared to the BoB data. We attribute this difference to the inherently different nature of questions and answers across the two data sets; as shown in Section 2., BoB answers consist of highly grammatical English sentences, while Bertomeu questions and answers tend to consist of long lists of dates or proper names.

Dialogue and Discourse features Table 11 shows how the different dialogue data sets differ in terms of our Dialogue and Discourse features introduced in Section 3.2.. The percentages in the table represent the proportions of the data sets for which the respective features hold (i.e., evaluate to true, or to one of the four `center.Transition` values).
Regarding dialogue coherence in terms of the Centers, we make two observations from Table 11. Firstly, we note that compared to the realistic IQA data from the BoB data set, the Bertomeu data exhibit much lower counts of dialogue snippets where the `center.Forward` feature holds, i.e., where there is entity-based continuity between Q_1 and A_2. We attribute this difference to the typically list-like structure of A_2 in the Bertomeu data; the large difference in proportions indicates some unnatural property of the Bertomeu data. As for the Centering-Theory-based transitions between Q_1 and Q_2 described by the `center.Transition` feature, we note that the TREC data exhibit a rather large proportion of *continue* transitions; the numbers suggest a closer structural similarity of

	BoB	TREC'01	TREC'04	Bertomeu
Dialogues (= QA sessions)	1,161[a]	10	64	33
Number Q_2s (= nr. snippets)	1,522	32	221	1,052
Mean utterances per dialogue/QA session	3.86	4.2	4.47	66.2
Mean Q length (words)	4.4	7.7	6.0	8.8
Mean A length (words)	26	–	–	118

[a] 1,161 (or 66%) from a total of 1,765 dialogue sessions contained at least one FU Q.

Table 5: Quantitative measures of data sets

Nr. of user questions in dialogue	1	2	3	4	5	6	>6
Nr. of dialogues (tot.: 1,765)	604	313	246	160	112	73	257
Proportion of tot. dialogues	34.2%	17.7%	13.9%	9.1%	6.3%	4.1%	14.6%

Table 6: Counts and proportions of BoB dialogues along numbers of user questions

the TREC data to the topic continuation (TC) subset of the BoB data. This supports our observation that the questions in TREC cannot be taken to represent real user questions in IQA dialogues.

Finally, looking at the action-based coherence features in the last three rows of Table 11, the Bertomeu data show similar feature proportions as the BoB Topic Continuation (TC) data set, but rather unlike the complete BoB data set. This is another sign that the Bertomeu data do not represent naturally occurring topic shifting behavior between user questions, but rather seem to exhibit Topic Continuation properties.

5. Conclusion

We believe the IQA research community could benefit from a new evaluation campaign in the style of the TREC QA context track, but resolving its two shortcomings: the artificiality of user questions, and the lack of preceding system answers on which FU Qs might build. We would hope and expect such a campaign to give rise to a new wave of research in the area of discourse and context modeling, which would aim to improve an IQA system's ability to answer user FU Qs. In this paper, we have introduced and described our collection of BoB IQA dialogue data, and have shown how these data compare to other relevant data sets.

In Section 3. we introduced several measures for quantifying relations between utterances in IQA dialogue snippets, based on either string similarity, or different theories of Dialogue and Discourse coherence. We claim that these methods provide important insights into the inter-utterance structure of IQA dialogue data, and allowed us to point out relevant differences between the realistic BoB data set and two less natural data collections. The goal for a new evaluation campaign should be to provide a large set of IQA dialogues that resemble realistic data in different aspects, such as the features and measures we have introduced in this paper. We have introduced the set of BoB IQA dialogues as our attempt to provide such a data set to the research community.

In Section 4. we used the above-mentioned inter-utterance measures to pinpoint relevant differences between the data sets. However, we started by exploring differences that became evident already through the comparison of some quantitative measures. First of all, real users in an IQA setting do ask FU Qs: in the case of BoB, two thirds of all IQA sessions contained at least two user questions. As opposed to artificially collected user questions, real user questions tend to be relatively simple and short, with an average word length of 4.4 words. Comparing this average to the same measure calculated from the query logs of a commercial web search engine, which was 2.35 in the year 1998 (Silverstein et al., 1998), we note that realistic IQA user questions fare somewhere in between web search engine queries and questions from the more artificial IQA data sets we have analyzed here. Interesting steps for further research would be to analyze and compare web search query logs using the different measures that we proposed here, and to see if over the course of the last decade web search queries might have evolved towards longer, and maybe also more contextually related queries.

From our analysis of real IQA interaction logs regarding the occurrence of anaphora in FU Qs, we have indications that the previous system answer (A_1) plays an important role; the number of detected anaphora in the FU Q increased by 61% (relative) when previous system answers were considered as the possible location for antecedents. We see this contextual dependency of FU Qs as an indication that users do take the answers to their previous question into account when formulating a FU Q; it is thus essential to consider this fact also in a new IQA evaluation campaign that goes beyond the context questions task, from which the TREC questions described in this paper are taken. Further research should investigate how to treat different kinds of web search engine results as system answers, and explore to what extent FU Qs refer to such previous search results in a way similar to more traditional IQA systems.

We believe that realistic IQA dialogues like the BoB data described in this paper can serve as a basis for studying, modeling and predicting user topic shifting behavior, particularly with methods based on Machine Learning (e.g., (Kirschner et al., 2009)). Such a study is not possible using artificial IQA data, because even in the case of data originating from a Wizard-of-Oz experiment such as Bertomeu, topic shifts will be to a large extent determined by the user's particular task when conducting the experiment. On the other hand, empirical and supervised Machine Learning-based approaches are facilitated by the easy availability of rather large collections of realistic IQA dialogue data, e.g., in the form of dialogue snippets, as we have proposed in this paper. We would hope to see further realistic IQA dialogue collection efforts, possibly in other languages or domains, and the coordinated release of all resulting dialogue data sets to the IQA research community.

6. References

Raffaella Bernardi, Manuel Kirschner, and Zorana Ratkovic. 2010. Context fusion: The role of discourse structure and centering theory. In *Proceedings of LREC 2010*, Malta.

Nuria Bertomeu. 2008. *A Memory and Attention-Based Approach to Fragment Resolution and its Application in a Question Answering System*. Ph.D. thesis, Department of Computational Linguistics, Saarland University.

Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. 1987. A centering approach to pronouns. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 155–162, Stanford, California.

John Burger, Claire Cardie, Vinay Chaudhri, Robert Gaizauskas, Sanda Harabagiu, David Israel, Christian Jacquemin, Chin-Yew Lin, Steve Maiorano, George Miller, Dan Moldovan, Bill Ogden, John Prager, Ellen Riloff, Amit Singhal, Rohini Shrihari, Tomek Strzalkowski, Ellen Voorhees, and Ralph Weishede. 2000. Issues, tasks and program structures to roadmap research in question & answering (Q&A).

Joyce Y. Chai and Rong Jin. 2004. Discourse structure for context question answering. In *Proc. of the HLT-NAACL 2004 Workshop on Pragmatics in Question Answering*, Boston, MA.

Hoa Trang Dang, Diane Kelly, and Jimmy Lin. 2007. Overview of the TREC 2007 question answering track. In *Proc. of the 16th Text REtrieval Conference*.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of LREC 2006*, Genoa, Italy.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Mijail Kabadjov. 2007. *A Comprehensive Evaluation of Anaphora Resolution and Discourse-new Classification*. Ph.D. thesis, University of Essex.

Diane Kelly and Jimmy Lin. 2007. Overview of the trec 2006 ciqa task. *SIGIR Forum*, 41(1):107–116.

Diane Kelly, Paul B. Kantor, Emile L. Morse, Jean Scholtz, and Ying Sun. 2006. User-centered evaluation of interactive question answering systems. In *Proc. of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 49–56, New York, NY.

Manuel Kirschner, Raffaella Bernardi, Marco Baroni, and Le Thanh Dinh. 2009. Analyzing Interactive QA dialogues using Logistic Regression Models. In *Proc. of AI*IA*, Reggio Emilia, Italy.

Manuel Kirschner. 2010. *The Structure of Real User-System Dialogues in Interactive Question Answering*. Ph.D. thesis, Free University of Bozen-Bolzano, Italy.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan.

Mark T. Maybury, editor. 2003. *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA*. AAAI Press.

Ruslan Mitkov, Branimir K. Boguraev, John I. Tait, and Martha Palmer, editors. 2009. *Journal of Natural Language Engineering. Special Issue on Interactive Question Answering*, volume 15. Cambridge University Press.

Massimo Poesio and Mijail Kabadjov. 2004. A general-purpose, off-the-shelf anaphora resolition module: Implementation and preliminary evaluation. In *Proc. of the 4th International Conference on Language Resources And Evaluation (LREC)*, Lisbon, Portugal.

Zorana Ratkovic. 2009. Deep analysis in iqa: evaluation on real users' dialogues. Master's thesis, European Masters Program in Language and Communication Technologies.

Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. 1998. Analysis of a very large altavista query log. Technical Report 14, Compaq Systems Research Centre, Palo Alto, CA.

Tomek Strzalkowski and Sanda Harabagiu, editors. 2006. *Advances in Open Domain Question Answering*, volume 32 of *Text, Speech and Language Technology*. Springer Netherlands.

Tomek Strzalkowski. 2006. The future: Interactive, collaborative information systems. Slides presented at HLT-NAACL 2006 Workshop on Interactive Question Answering.

Mingyu Sun and Joyce Y. Chai. 2007. Discourse processing for context question answering based on linguistic knowledge. *Know.-Based Syst.*, 20(6):511–526.

Ellen M. Voorhees. 2001. Overview of the TREC 2001 question answering track. In *Proc. of the 10th Text REtrieval Conference*.

Ellen M. Voorhees. 2004. Overview of the TREC 2004 question answering track. In *Proc. of the 13th Text REtrieval Conference*.

Fan Yang, Junlan Feng, and Giuseppe Di Fabbrizio. 2006. A data driven approach to relevancy recognition for contextual question answering. In *Proc. of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 33–40, New York City, NY.

	BoB	TREC'01	TREC'04	Bertomeu
Number Q_2s (= nr. snippets)	1,522	32	221	1,052
pers-pro (GuiTAR)	71 (5%)	5 (16%)	48 (22%)	110 (10%)
pn (GuiTAR)	115 (8%)	2 (6%)	4 (2%)	159 (15%)
the-np (GuiTAR)	9 (1%)	0	5 (2%)	22 (2%)
the-pn (GuiTAR)	0	1 (3%)	0	4 (0%)
Anaphora total (GuiTAR)	195 (13%)	8 (25%)	57 (26%)	295 (28%)

Table 7: Comparison of **automatic anaphora counts** across data sets, without considering previous answer (A_1). Percentages are out of total Q_2s.

	Anaphora in Q_2, considering A_1	Anaphora in Q_2, **without** considering A_1	Relative change (additional anaphora through A_1)
Personal pronoun (pers-pro)	130 (9% of 1,522)	71 (5%)	+ 59 (+ 83%)
Proper noun (pn)	140 (9%)	115 (8%)	+ 25 (+ 22%)
Definite NP (the-np)	43 (3%)	9 (1%)	+ 34 (+ 378%)
Anaphora total	313 (21%)	195 (13%)	+ 118 (+ 61%)

Table 8: Automatic anaphora counts (using GuiTAR) in 1,522 BoB Q_2s, with/without **considering previous answer** (A_1)

	TREC'01	TREC'04
Number Q_2s (= nr. snippets)	32	221
Anaphora total (automatic)	8 (25%)	57 (26%)
Anaphora total (manual)	22 (69%)	173 (78%)

Table 9: Comparing automatic anaphora counts (using GuiTAR) with **manual anaphora counts** in TREC data

	BoB	BoB, TC Q_2s only	TREC'01	TREC'04	Bertomeu
$Q_1.A_1$	0.08	0.12	–	–	0.09
$Q_1.Q_2$	0.24	0.39	0.25	0.31	0.30
$Q_1.A_2$	0.08	0.12	–	–	0.05
$A_1.Q_2$	0.08	0.09	–	–	0.06
$A_1.A_2$	0.14	0.16	–	–	0.14
$Q_2.A_2$	0.18	0.17	–	–	0.09

Table 10: Comparison of average inter-utterance **string similarities** across data sets

	BoB	BoB, TC Q_2s only	TREC'01	TREC'04	Bertomeu
Number Q_2s (= nr. snippets)	1,522	250	32	221	1,052
`center.Reference`($Q_2 \rightarrow A_2$)	3%	4%	–	–	3%
`center.Forward`($Q_1 \rightarrow A_2$)	39%	53%	–	–	15%
`center.Transition`($Q_1 \rightarrow Q_2$): continue	8%	17%	16%	40%	14%
`center.Transition`($Q_1 \rightarrow Q_2$): retain	2%	3%	0%	1%	2%
`center.Transition`($Q_1 \rightarrow Q_2$): smoothShift	1%	2%	0%	3%	1%
`center.Transition`($Q_1 \rightarrow Q_2$): roughShift	89%	78%	84%	55%	82%
`ConstraintRefinement`($Q_1 \rightarrow Q_2$)	4%	7%	3%	8%	7%
`ParticipantShift`($Q_1 \rightarrow Q_2$)	4%	8%	3%	6%	9%
`TopicExploration`($Q_1 \rightarrow Q_2$)	28%	48%	28%	52%	42%

Table 11: Comparison of **Dialogue and Discourse features** across data sets in proportions of all Q_2s (= snippets). Without considering previous answer A_1.

A Comparative Analysis: QA Evaluation Questions versus Real-world Queries

Johannes Leveling

Centre for Next Generation Localisation
School of Computing
Dublin City University
Dublin 9, Ireland

Abstract

This paper presents a comparative analysis of user queries to a web search engine, questions to a Q&A service (`answers.com`), and questions employed in question answering (QA) evaluations at TREC and CLEF. The analysis shows that user queries to search engines contain mostly content words (i.e. keywords) but lack structure words (i.e. stopwords) and capitalization. Thus, they resemble natural language input after case folding and stopword removal. In contrast, topics for QA evaluation and questions to `answers.com` mainly consist of fully capitalized and syntactically well-formed questions. Classification experiments using a naïve Bayes classifier show that stopwords play an important role in determining the expected answer type. A classification based on stopwords is considerably more accurate (47.5% accuracy) than a classification based on all query words (40.1% accuracy) or on content words (33.9% accuracy). To simulate user input, questions are preprocessed by case folding and stopword removal. Additional classification experiments aim at reconstructing the syntactic wh-word frame of a question, i.e. the embedding of the interrogative word. Results indicate that this part of questions can be reconstructed with moderate accuracy (25.7%), but for a classification problem with a much larger number of classes compared to classifying queries by expected answer type (2096 classes vs. 130 classes). Furthermore, eliminating stopwords can lead to multiple reconstructed questions with a different or with the opposite meaning (e.g. if negations or temporal restrictions are included). In conclusion, question reconstruction from short user queries can be seen as a new realistic evaluation challenge for QA systems.

1. Introduction

User queries to search engines usually consist of 2-3 words (Spink et al., 2001; Teevan et al., 2006) and rarely are formulated as full sentences or questions (Ozmutlu et al., 2003). Query processing for information retrieval (IR) systems typically involves transforming the original user query by successively applying case folding, stopword removal, and stemming. Thus, user input to search engines already resembles results from query processing (as illustrated in Table 1) in that it typically lacks capitalization and stopwords, but still contains full word forms. Provided that most users are accustomed to web search engines but not familiar with QA systems, or that users mistake QA systems for information retrieval systems, they will try to formulate requests to QA systems as short keyword queries.

A comparative analysis of queries and questions investigates differences between queries (i.e. user input to search engines or preprocessed questions after case folding and stopwords removal) and questions (i.e. full natural language requests to QA systems). Different aspects of queries and questions are analyzed, including average length, case information, occurrence of stems and full word forms, wh-words (interrogative words), and sentence delimiters. The comparison aims at finding differences and similarities between QA questions and real-world queries.

The analysis demonstrates that much of the information present in full natural language questions is missing in short user queries. Thus, natural language processing tasks for QA, such as determining the expected answer type, cannot be performed as reliable as for full questions. However, a simple classification experiment illustrates that part of a question (the syntactic frame including the wh-word) can be correctly generated for 25.7% of the queries, despite of problems such as ambiguous queries.

2. Related Work

Brown and Coden describe an approach to reconstruct capitalization in text, trained on news stories (Brown and Coden, 2002). Their system assumes full punctuation of text. They infer that any word that does not appear in their capitalization dictionary (i.e. out-of-vocabulary) is most likely a proper noun and should be capitalized. Their best approach is based on capitalization dictionaries, phrases and other context information such as punctuation and achieves a precision of 90.3% and recall of 88.2%.

Gravano, Jansche et al. try to recover capitalization and punctuation in automatic speech transcripts using an n-gram language model (Gravano et al., 2009). Experiments are based on 1989 Wall Street Journal corpus and Broadcast News and show that using larger training corpora improves performance, but increasing the gram size from 3 to 6 does not. They assume that at most one punctuation symbol can occur between two words and use a limited set of punctuation characters (e.g. quotation marks are excluded).

Edmonds investigates lexical choice (Edmonds, 1997). He uses lexical co-occurrence networks based on mutual information and significance scores to fill word gaps with the most typical synonym. The system was trained on the part-of-speech tagged 1989 Wall Street Journal. Results show that including second-order co-occurrences improve performance of the system.

In summary, the reconstruction of punctuation and capitalization has been researched in automatic speech recognition and machine translation (MT) (Brown and Coden, 2002; Gravano et al., 2009; Huang and Zweig, 2002), but typically focuses on processing full text (e.g. news stories or automatic speech transcripts) instead of short queries. In addition, most of the research so far has ignored that stopwords and interrogative words are much more important in QA than in IR. For example, the wh-word (inter-

rogative word) is an important feature in determining the expected answer type (EAT) of a question and *full* natural language questions are required if a QA system builds on deep syntactic-semantic parsing of questions or on other complex NLP methods. However, short user queries seldom contain interrogative words (cf. Table 2).
Approaches to finding questions describing information needs are also realized in systems for FAQ (frequently asked questions) search. Instead of trying to reconstruct questions from user input, the input is compared to questions in a question collection to find similar ones. Commercial solutions such as `q-go.com`[1] focus on closed domains (typically a single web site) which limits the type and number of possible questions. Using syntactic and morphologic information, user queries are mapped to sets of possible questions and multiple alternatives are presented to the user. For the open domain, these approaches would require a huge number of previously entered questions.
Spink, Wolfram et al. analyzed Excite query logs with more than 1 million queries and found that the average query length is 2.4 words (Spink et al., 2001). They also find that among the top 75 frequent terms used in queries, many are no-content terms (e.g. *'and'*, *'of'*, *'the'*, *'in'*, *'for'*, *'+'*, *'on'*, *'to'*, *'or'*, *'&'*, *'a'*). Similarly, Teevan, Adar et al. found that the average query length to search engines is 2.7 words (Teevan et al., 2006).
Leveling manually annotated the MultiNet Bibliographic Query Corpus, which consists of 12.946 user questions to a German natural language interface (NLI) to information providers on the internet (Leveling, 2006). 28.2% of the annotated queries contain some form of error, including wrong capitalization and spelling errors. Also, users were observed to formulate longer queries compared to search engines once they found out that the system can process full sentences and questions. As in web search, users of the NLI often enter one or two-word queries, confusing the NLI with a keyword-based search engine. Some other users entered much longer requests, similar to a dialogue with a human librarian. In contrast to web search, the natural language questions to this NLI contain 7.58 words on average. Using the query collection as a test set, structured database queries could be generated even for short queries or malformed information requests, using an automatic classification of terms. This approach increases the number of correctly transformed queries by about 30%.
Clearly, there is a gap between real-world user queries and questions used in evaluation campaigns such as TREC[2] or CLEF[3]. For example, questions and queries used in evaluation campaigns are typically grammatically well-formed, but user queries (e.g. in search engine logs or mailing lists) are not necessarily.

3. Analysis of questions and queries

3.1. Corpora

Six corpora containing queries, questions, and sentences were analyzed.

1. The question collection from Webclopedia (Hovy et al., 2000), a question answering system which has been evaluated at TREC. This collection originates from `answers.com`[4], a commercial Q&A service, providing answers to user questions. The hierarchical Webclopedia question typology (Hovy et al., 2002) was developed on an annotated set of questions from this corpus.[5]
2. The Excite log (Excite) of user queries, as distributed in Pig[6]. Pig is a software tool for analysis of large data sets and query logs and is being developed as an open source project under the Apache Software Foundation.
3. The Wikipedia article names[7] (titles) of the English Wikipedia.
4. The English 1 million sentence corpus from the Leipzig corpus collection[8], which contains samples from newspaper articles (EN1M). This resource originates from newspaper articles and has been collected for co-occurrence analysis (Quasthoff et al., 2006).
5. More than 2300 questions from the main TREC question answering track (see, for example (Voorhees and Tice, 2000)).
6. The combined English questions from the multi-6, multi-8, and multi-9 corpora (short: multi-X). Parts of these corpora have been used for official QA evaluation at CLEF QA 2003-2006 (see, for example (Magnini et al., 2006)).

The following processing steps were carried out to determine if a word form is a base form (stem): The Porter stemmer (Porter, 1980) was applied to the words in the text. If the stemmed result is equal to the input, the word is presumed to be the base form. Note that this approach is only a heuristic, because overstemming or understemming might produce results different from a correct base form. Also, stemming may result in words resembling stopwords.
For the data analysis, all text was tokenized by splitting at special characters (e.g. underscore, ampersand, brackets, the at-sign, etc.) and punctuation symbols (i.e. *','*, *';'*, *'?'*, *'!'*, *':'*, *'.'*, *'-'*). Following this tokenization method, URLs are not recognized as a single token, but are split into several tokens, including words (e.g. *'http'* and special characters (e.g. *':'*).

3.2. Queries and Questions

Results of the analysis of this data are shown in Table 1, confirming that the average length of user queries (in column Excite) is 2–3 words (Spink et al., 2001; Teevan et al., 2006). In addition, the following observations have been made: User queries (Excite) rarely contain stopwords,

[1] `http://www.q-go.com/`
[2] `http://trec.nist.gov/`
[3] `http://www.clef-campaign.org/`
[4] `http://www.answers.com/`
[5] `http://www.isi.edu/natural-language/projects/webclopedia/Taxonomy-data/WH_Analysis.html`
[6] `http://hadoop.apache.org/pig/`
[7] `http://dumps.wikimedia.org/`
[8] `http://corpora.uni-leipzig.de/`

punctuation symbols, or uppercase words in comparison to the full sentence corpus. Special characters (e.g. quotation marks or *'-'*) often indicate web queries with special syntax, e.g. a phrase search or exclusion of terms. Wikipedia article names contain an even higher proportion of capitalized words, but capitalization occurs in expected places, e.g. at the beginning of sentences. Thus, the percentage of capitalized words is much higher in comparison with corpora containing full sentences. Users still enter full words forms as query terms for a web search (52.9% stems, 47.1% non-stems for Excite), assuming that the search engine will handle morphological variation or exact matching of query terms. Contrary to expectation, short user queries still contain full word forms, but the corresponding values are much higher for full sentences, for evaluation questions, and for questions to `answers.com` (67.1%-76.6%).

The analysis shows that case information and stopwords are mostly missing in web queries. This is not the case for questions from evaluation benchmarks such as TREC QA or QA@CLEF, and for questions to `answers.com`, where queries are typically well-formed, because syntax and orthography are expected to be correct or because malformed questions may prove expensive.

The following conclusions can be drawn from the analysis: Real-world user queries are short, contain few stopwords and lack capitalization. Thus, natural language processing (NLP) tasks such as part-of-speech (PoS) tagging, named entity recognition (NER) and classification, or parsing of user queries will most likely fail. Typically, these tasks are solved by approaches employing statistical methods trained on large corpora consisting of syntactically correct sentences. For example, in contrast to full text, short queries will contain different n-grams and no real syntactic structure because stopwords are missing. Furthermore, missing capitalization makes named entity recognition more difficult because proper nouns are not capitalized. In the EN1M corpus, only a small fraction of all sentences are questions. That means that part-of-speech tagging for queries and questions will be difficult even if a tagger is trained exclusively on questions extracted from a larger corpus (because of the smaller training set). Annotated corpora consisting of user queries are still too small to be useful in practice or are not yet available to the research community.

While the query logs provided with the Pig tool may seem a bit dated, user queries for web search engines do not seem to change over long periods of time. Silvestri (Silvestri, 2010) presents comparative statistics for query logs from Altavista and Excite from 1997-2002 which are based on experiments by Spink et al. (Spink et al., 2002). He observes that query behaviour has not changed from a statistical point of view over a period of four years and shows that query characteristics such as the number of terms per query vary only slightly or remain unchanged over time.

3.3. Identifying questions in query logs

The following characters were defined as sentence delimiters: *'?'*, *'!'*, *'.'*, *';'*, *""*, *')'*, *']'*, and *'}'*. Interrogative words of the following types were considered as wh-words to identify questions (here denoted by tags from the CLAWS tagset, (Garside et al., 1997)): AVQ (wh-adverb, e.g. *'when'*, *'how'*, *'why'*), DTQ (wh-determiner, e.g. *'whose'*, *'which'*), and PNQ (wh-pronoun, e.g. *'who'*, *'whoever'*).

The list of wh-words was compiled from a tagged subset of the British National Corpus (BNC). Therefore, some spelling mistakes and contracted forms of interrogative words are also included. Table 2 shows the top 10 most frequent words, wh-words, and sentence delimiters for the corpora.

The use of question marks as sentence delimiters indicates that most topics in QA evaluation and in questions from `answers.com` constitute proper questions. In some cases, the entries are multiple choice queries and answers were provided together with the question. In these cases, the user input does not end with a question mark.

Many questions to `answers.com` take the form of a natural language question, but some requests are formulated as imperative sentences or simple statements. In rare cases, imperative forms of verbs indicate a request for information (e.g. *"give ..."*, *"find ..."*, *"list ..."*), i.e. the queries should end with an exclamation mark (but typically do not).

The Excite log contains several special characters among the top-ten (most frequent) terms (e.g. *':'*, *'/'*, *'+'*, and *'.'*). These are artefacts from splitting up URLs into several tokens and from special operators used in search engines to denote the inclusion or exclusion of terms. Thus, these special characters appear among the top frequency terms.

Another experiment aims at identifying natural language questions in the Excite query log by looking for wh-words in the first five terms of a question and for a question mark in the last three tokens. If any of these is found, the entry is flagged as a potential question. In the Excite log with about 1 million queries, less than 5000 entries were found to be questions. The most frequent type of question observed are *"how to"*-questions (e.g. *"how to write a resume"*). This type of question is difficult to answer even for automatic QA systems. However, this asserts that some users seem to be looking for answers to this kind of question (which can be answered reasonably well by providing a web page).

3.4. Duplicate questions and ambiguity

After case folding and stopword removal have been applied to the questions from `answers.com`, duplicate questions were identified and their annotated classes (EAT) compared. To test if two queries are duplicates, they are represented as sets of content words Q_1 and Q_2. If $Q_1 / Q_2 = Q_2 / Q_1 = \emptyset$, the queries are regarded as duplicates, i.e. if they consist of the same content words. If two duplicates are tagged with different classes, this indicates either that a) the annotation was inconsistent or b) a possible question reconstruction is ambiguous, because more than one syntactic frame can be generated for queries with the same content words.

For example, the single word input *"berlin"* might be transformed into *"Where is Berlin?"* or *"What do you know about Berlin?"*; and vice versa: both of the latter queries are reduced to the single word query *"berlin"* after preprocessing. The detection of duplicate queries shows that 773 questions out of 22223 (3.5%) are duplicates.

Alternative interpretations of short user queries (ambiguity)

Table 1: Analysis of English corpora and topics.

		Excite	EN1M	Wikipedia	TREC	multi-X	answers.com
type		User queries	Sentences	Titles	Questions	Questions	Questions
entries		0.94M	1M	7.18M	2393	2580	35287
tokens		2.45M	25.1M	23.3M	20381	23238	381482
avg. length		2.6	25.1	3.2	8.52	9.00	10.81
uppercase	[%]	0.7	13.8	66.6	27.4	31.4	23.6
lowercase	[%]	81.8	70.6	17.7	58.0	53.6	61.7
numeric	[%]	4.9	2.1	2.4	0.4	1.7	1.1
punctuation	[%]	6.8	11.2	5.3	13.8	13.1	13.1
special	[%]	5.8	2.3	7.9	0.2	0.3	0.5
stopwords	[%]	7.8	49.0	11.7	53.4	51.9	53.3
non-stopwords	[%]	92.2	51.0	88.3	46.6	48.1	46.7
stems	[%]	52.9	28.5	8.3	30.6	23.4	32.9
non-stems	[%]	47.1	71.5	91.7	69.4	76.6	67.1

might be resolved by a simple popularity vote, i.e. using web search engines to obtain the frequencies of different questions via an exact search and selecting the most frequent (i.e. the most popular) alternative. However, this approach will not work for questions aiming at recent events because web search engines have to be updated regularly and modified content is indexed and available with some delay. Furthermore, users may actually mean the less popular interpretation because otherwise a simple web search might suffice to fulfil the information need.

In conclusion, generating a single question from short user input may not increase user satisfaction if the question can not be generated correctly. Instead, different questions should be suggested to the user for selection. Selecting full questions from a set of alternatives shown in the QA system interface might also help alter the user behaviour faster if the user learns that a QA system accepts or expects full natural language input.

4. Question Reconstruction

4.1. A simple approach to reconstructing the syntactic wh-word frame

To obtain a full natural language request for a QA system from a user query, the syntactic wh-word frame has to be created. The wh-word frame is defined as the longest stopword sequence at the start of a question, which is used as a class label in the following experiments. For example, the query *"capital Ethiopia"* is missing the wh-word frame *"what is the"*, the preposition *"of"* between *"capital"* and *"Ethiopia"*, and a trailing question mark to form the question *"What is the capital of Ethiopia?"*. There may be more than one correct wh-word frame, e.g. *"boston tea party"* could mean *"When was the Boston Tea Party?"*, *"Where was the Boston Tea Party?"*, or even – assuming a changed word order – *"Which party in Boston makes tea?"*. The corresponding wh-word frames for these examples are *"when was"*, *"where was"*, and *"which"*.

A trivial method for question reconstruction is to add a single generic syntactic frame *"Find information about ..."* and a single type of stopword (i.e. *'AND'*) to the user query to form a full request. This default approach works reasonably well for user queries containing a single word or proper nouns. Multiple content words can be connected by adding the word *'AND'* between them. For example, the query *"violence schools"* would be transformed into *"Find information about violence and schools"*.

However, this approach creates only general requests for *information* on a topic, not specific questions. The EAT for this type of question is overly generic and all reconstructed questions will be associated with the same EAT, i.e. this type of questions would be more suitable for a web search engine, not for a QA system. In a real-world QA scenario, users may be more interested in specific aspects of a topic. Furthermore, adding *'AND'* between all words also breaks up multi-word expressions and multi-word names (e.g. *'AND'* should not be inserted between *"New"* and *"York"*). Hence, a non-trivial solution for question reconstruction is needed.

In this paper, question reconstruction focuses on finding the wh-word frame given a case-folded query after stopword removal (which simulates the user query). This task seems to be similar to finding the expected answer type for QA, but there are some important differences: In contrast to finding the EAT, PoS information, named entity tags, and even capitalization information is not reliable or available for short queries. For instance, part-of-speech taggers are typically trained on an annotated corpus with full sentences (which contains fully capitalized words and stopwords, see EN1M in Table 1). Tagging will not be accurate if properties of the input (user queries) do not match properties of the training data. In addition, the word ordering may be different from the order in the final (or intended) question.

Table 3 shows results of classification experiments using a naïve Bayes classifier to determine the expected answer type and the wh-word frame. The training data consists of the questions from `answers.com`, together with their annotated expected answer type (EAT, called *qtarget* in Webclopedia) from the taxonomy used in the Webclopedia QA system. The question collection from `answers.com` was processed by filtering out entries missing an EAT and cor-

Table 2: Analysis of English corpora and topics.

	Excite	EN1M	Wikipedia	TREC	multi-X	answers.com
Top-10 words	*’NUM’*	*’the’*	*’(’*	*’the’*	*’the’*	*’the’*
	’+’	*’,?’*	*’NUM’*	*’what’*	*’is’*	*’what’*
	””	*’of’*	*’of’*	*’is’*	*’what’*	*’is’*
	’/’	*’to’*	*’-’*	*’of’*	*’in’*	*’of’*
	’and’	*’NUM’*	*’,’*	*’in’*	*’of’*	*’in’*
	’-’	*’a’*	*’the’*	*’was’*	*’who’*	*’a’*
	’of’	*’in’*	*’in’*	*’who’*	*’was’*	*’was’*
	’,’	*’and’*	*’and’*	*’how’*	*’which’*	*’who’*
	’the’	*’-’*	*’List’*	*’did’*	*’NUM’*	*’,’*
	’:’	*””*	*’de’*	*’a’*	*’did’*	*’NUM’*
Wh-words						
’what’	472	16322	3315	1286	873	18422
’how’	1453	9090	2112	304	312	2929
’when’	67	29968	1666	202	179	515
’who’	326	47291	4203	297	549	4748
’where’	276	12432	1239	161	185	1494
’which’	33	48545	288	63	385	927
’what’s’	22	836	555	23	3	2255
’why’	94	3161	740	8	1	846
’whom’	2	1324	77	6	5	42
’where’s’	3	34	163	1	0	54
’who’s’	45	356	398	1	2	204
’whose’	1	4208	93	1	6	159
’howe’	38	88	597	0	0	0
’whatsoever’	10	92	0	0	0	0
’whatever’	10	917	213	0	0	6
’wherever’	1	135	33	0	0	0
’how’s’	0	7	0	0	0	1
other	942095	825194	7165622	39	80	2685
Delimiters						
’?’	391	5218	4048	2353	2486	33938
’.’	4471	970144	43158	35	90	523
’!’	45	250	8959	0	0	8
””	44929	22054	7698	3	1	315
”’	1601	2334	37339	0	0	8
’:’	264	0	83	0	0	0
’)’	151	0	743051	0	0	22
’]’	3	0	0	0	0	1
other	893093	0	6336978	1	3	472

recting spelling errors in the class labels. Disjunctions of EAT were resolved by using only the first question annotation. There are 130 EATs used in the annotated questions. The data was divided into a training set containing 22223 questions (about 90%) and a test set containing the remaining 2222 instances.

Using lower case words after stopword removal as classification features, a naïve Bayes classifier was trained on the annotated questions. For the first three classification experiments, the class to be determined is the EAT. The annotated questions are associated with 130 classes, which correspond to the fine-grained hierarchical taxonomy of answer types used in Webclopedia (Hovy et al., 2002; Hovy et al., 2000). For the final experiment, the stopword sequence at the beginning of the original question (the wh-word frame) was used as a class label. This type of classification will automatically determine the syntactic wh-word frame of a question via the class label.

The results for the first three experiments seem to indicate that correct classification may rely largely on present stopwords in the question. The classification experiment using stopwords only as features achieves a much higher accuracy (47.5%) than EAT classification based on all words (40.1%) and on content words only (33.9%). Some natural language processing tasks for QA rely on stopwords (e.g. n-gram models), and if stopwords or other information is missing from input to a QA system (as was observed for web search queries), NLP processing will likely show degraded performance compared to applying the same method on full sentences or questions.

The final classification experiment investigated if the syntactic frame of a question can be generated for short user input. The user input (which is to be classified) is simulated by case folding full natural language questions to lower case and removing all stopwords. For this experiment, a much lower accuracy has been observed, compared to classification of EAT. However, the number of classes corresponds to the different surface realizations of the wh-word frame in a question (2096 classes compared to only 130 classes for classifying the EAT), which makes a classification much more difficult. Therefore, the results of this baseline experiment seem promising: in almost 26% of all simulated queries, the correct wh-word frame can be generated to form a full natural language question from only partial information.
Improvements for this approach are obvious, but may be difficult to realize: Using additional information such as the part-of-speech or named entity class of words in the input will help to improve accuracy. However, this information may not be obtained with high accuracy from lower case keywords. There is no need to exactly reproduce the wh-word frame of the original question. There may be different paraphrases of the same question expressing the same meaning. Currently, paraphrases of the wh-word frame are counted as errors. Recovering capitalization, missing stopwords (and possibly full word forms) will help to create a full natural language question which can be used in QA systems as a replacement for the terse original user input. However, the *full* query reconstruction is beyond the scope of this paper.

4.2. Discussion

Do user queries contain enough information to reconstruct a full natural language question? There are many problems making the task of question reconstruction a difficult one. So far, little research has investigated the problems that arise from reconstructing full natural language questions from partial information.

Ambiguous input. The user input *"bush fire sydney"* (after stopword removal and case folding) can be transformed into different questions, e.g. *"Will Bush fire Sydney?"* and *"Are there any Bush fires near Sydney?"*. Note that these questions have a different EAT and will be treated differently by QA systems, i.e. QA systems are expected to generate different answers. Without additional knowledge on the user, domain, or document collection, this ambiguity cannot be resolved.

Question paraphrases. There often are several possible alternatives which can be reconstructed for a given input. For example the query *"food lions"* could imply the intended question *"How much food do lions eat?"* or *"What food do lions eat?"*. In both cases, a verb which is closely related to the query topic has to be added (*'eat'*).

Word class conversion. User queries often contain nouns instead of adjectives or adverbs. For example, the query *"height Bruce Willis"* may have to be reformulated as *"How tall is Bruce Willis?"* instead of *"What is the height of Bruce Willis?"*.

Question types (overspecific or underspecific). Yes-no questions are mainly implied by stopwords, e.g. *"Has it ever snowed in Miami, FL?"* or *"Is a cello larger than a viola?"* are difficult to create from the user input *"snow Miami"* and *"cello (larger) viola"*. Instead, a more specific question might be reconstructed, e.g. *"When did it snow in Miami, FL?"*. In contrast, users may also be interested in general information about a subject (e.g. *"What do you know about artificial intelligence?"*).

Converse, contrastive or negated meaning of resulting question. The proposed approach for query reconstruction will have some limitations inherited from traditional IR: negations and contrary meanings can be reconstructed accurately only for the most frequent cases and will have to be ignored otherwise. Similar problems arise from temporal or spatial restrictions (e.g. *"without work"* vs. *"with work"*; *"hotels in X"* vs. *"hotels outside of X"*; *"X after 1995"* and *"X before 1995"*).
Negation expressed by stopwords can not be recovered at all (e.g. *'non'*, *'no'*) if these words do not occur in a query. Users interested in exact or specific answers will likely try to explicitly express this and include some form of negation in their query.

Complex questions. One assumption that is often made is that user input consists of a single sentence. However, questions from `answers.com` show that sometimes the answer is already contained in the question (e.g. *"Is Bill Clinton a lefty or righty?"* , *"True or false: ..."*). The question *"Do people only use 10% of their brains? If so, why?"* also shows that user questions can be more complex. In this example, the assumption of a single query does not hold. Two questions are asked, which are associated with different answer types: Y:N (yes-no-question) and REASON (reason explanation).
Fortunately, most of these linguistic phenomena (e.g. negation) are outside the scope of state-of-the-art QA systems. In conclusion, a first approach at full question reconstruction should follow the principle of Occam's razor and assume that the simplest question that can be constructed is the question intended by the user.

5. Conclusions and Outlook

The major findings from the analysis and experiments described in this paper are: i) User queries to search engines lack capitalization and stopwords. In properties such as average length, they are most similar to questions after case folding and stopword removal. In contrast, questions to `answers.com` and questions in QA evaluations are mostly formulated as full natural language questions.
ii) Given that many users are accustomed to web search, but few know QA systems and their capabilities, user behaviour (and queries) for QA systems can be presumed to be similar to web search. Question reconstruction will help to improve the input to QA systems until users have adapted to QA systems. If this assumption is valid, questions from QA evaluation campaigns do not adequately represent the challenge of question answering in the "real world" (e.g. in mailing lists or web fora). Future QA research should include question reconstruction from short user queries, which poses

Table 3: Classification experiments for expected answer type and wh-word frame.

Experiment	Features	# Classes	Correct	Incorrect
EAT	all words	130	892 (40.1%)	1330 (59.9%)
EAT	stopwords	130	1055 (47.5%)	1167 (52.5%)
EAT	non-stopwords	130	754 (33.9%)	1468 (66.1%)
wh-word frame	non-stopwords	2096	570 (25.7%)	1652 (74.3%)

challenges including resolving the ambiguity of restored questions and handling negation.
iii) NLP tasks for short user queries will be more difficult, because in comparison to full natural language questions, important information for classification and other processing is missing.
Future work will include investigating part-of-speech tagging for short user queries to help constructing full questions and employing state-of-the-art approaches to classification (e.g. support vector machines).

6. Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (CNGL) project.

7. References

Eric W. Brown and Anni R. Coden. 2002. Capitalization recovery for text. In Anni R. Coden, Eric W. Brown, and Savitha Srinivasan, editors, *Information Retrieval Techniques for Speech Applications*, volume 2273 of *LNCS*, pages 11–22. Springer.

Philip Edmonds. 1997. Choosing the word most typical in context using a lexical co-occurrence network. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 507–509, Morristown, NJ, USA. ACL.

Roger Garside, Geoffrey Leech, and Anthony McEnery. 1997. *Corpus annotation: Linguistic Information from Computer Text Corpora*. Longman, London, New York.

Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009)*, pages 4741–4744. IEEE.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. 2000. Question answering in Webclopedia. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, pages 655–664.

Eduard Hovy, Ulf Hermjakob, and Deepak Ravichandran. 2002. A question/answer typology with surface text patterns. In *Proceedings of the second international conference on Human Language Technology Research*, pages 247–251, San Francisco, CA, USA. Morgan Kaufmann.

Jing Huang and Geoffrey Zweig. 2002. Maximum entropy model for punctuation annotation from speech. In *7th International Conference on Spoken Language Processing (ICSLP'02)*, pages 917–920.

Johannes Leveling. 2006. *Formale Interpretation von Nutzeranfragen für natürlichsprachliche Interfaces zu Informationsangeboten im Internet*. Der andere Verlag, Tönning, Germany.

Bernardo Magnini, Danilo Giampiccolo, Lili Aunimo, Christelle Ayache, Petya Osenova, Anselmo Peñas, Maarten de Rijke, Bogdan Sacaleanu, Diana Santos, and Richard Sutcliffe. 2006. The multilingual question answering track at CLEF. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odjik, and Daniel Tapias, editors, *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, pages 1156–1163, Genoa, Italy, 22-28 May 2006.

Seda Ozmutlu, Huseyin Cenk Ozmutlu, and Amanda Spink. 2003. Are people asking questions of general web search engines? *Online Information Review*, 27(6):396–406.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Uwe Quasthoff, Matthias Richter, and Chris Biemann. 2006. Corpus portal for search in monolingual corpora. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odjik, and Daniel Tapias, editors, *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, pages 1799–1802, Genoa, Italy, 22-28 May 2006.

Fabrizio Silvestri. 2010. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4(1-2):1–174.

Amanda Spink, Dietmar Wolfram, Major B. J. Jansen, and Tefko Saracevic. 2001. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3).

Amanda Spink, Bernard J. Jansen, Dietmar Wolfram, and Tefko Saracevic. 2002. From E-Sex to E-Commerce: Web search changes. *Computer*, 35(3):107–109.

Jaime Teevan, Eytan Adar, Rosie Jones, and Michael A. S. Potts. 2006. History repeats itself: Repeat queries in Yahoo's query logs. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 703–704. ACM.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, July 24-28, 2000*, pages 200–207. ACM.

Question Answering System Based on Community QA

Zhicheng Zheng, Yang Tang, Chong Long, Fan Bu, Xiaoyan Zhu

State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
{zhengzc04,tangyang9,randy2008,bufan0000}@gmail.com, zxy-dcs@tsinghua.edu.cn

Abstract

After a long period of research in factoid QA, such kind of questions has already been solved quite well. However, real users always concern on some more complicated questions such as "Why XXXX?" or "How XXXX?". These questions are difficult to retrieve answers directly from internet, but the community question answering services provide good resources to solve these questions. As cQA portals like Yahoo! Answers and Baidu Zhidao have attracted over hundreds of millions of questions, these questions can be treated as users' query log, and can help the QA systems understand the user's questions better. Common approaches focus on using information retrieval techniques in order to provide a ranked list of questions based on their similarity to the query. Due to the high variance of quality of questions and answers, users have to spend lots of time on finding the truly best answers from retrieved results. In this paper, we develop an answer retrieval and summarization system which directly provides an accurate and comprehensive answer summary besides a list of similar questions to user's query. To fully explore the information of questions and answers posted in the cQA, we adopt different strategies according to different situations. By this way, the system could output great answers to users' questions in practice.

1. Introduction

Retrieving precise information from internet becomes more and more important since the web data explodes. Question answering system aims to give an exact answer which is useful and relevant to user's query. There has been quite a long period of research in factoid QA driven by annual tracks at CLEF[1], TREC[2] and NTCIR[3]. Some systems in the tracks can answer simple factoid question with high accuracy. However such kind of questions is not usually from real users. Users often want to get answers for those more complicated question, such as "Why XXXX?" or "How XXXX?". These questions are difficult for following reasons: 1. It's quite hard to make computer understand what users want. 2. The unstructured internet data lacks of enough clear clues which indicate whether the text can be used to answer the question.

How to solve those questions is a new challenge for QA research. Community QA service is a quite useful resource for such research. In general, a cQA service has the following workflow. First, a question is posted by the user in the community and then others will answer the question. Usually there will be a best answer which is selected by the question poster or voting. The cQA portals like Yahoo! Answers and Baidu Zhidao have attracted over hundreds of millions of questions. All the questions can be treated as users' query logs. This can help the QA system understand the user's question more easily just like users' query logs help search engine understand users' query. When the QA system tries to solve a user's question, it may retrieve some similar questions from the cQA database. The answers to those similar questions will be good answer candidates to the user's question.

The unstructured web data lacks of enough clear clues, so it's not easy to get answer to user's question directly from internet. In the other hand, the answers in cQA all have corresponding questions. These questions are strong clues to indicate whether the answers should be used as answer to user's question. An intuitive way to answer user's question by using cQA resources is to find the most similar question to the user's question from cQA database, then use the best answer to the similar question as the answer to the user's question. However, this method will account some problems:

1. Sometimes there is no question in cQA database which is nearly same as the user's question. In such case, the answer getting from the most similar question will be ill.

2. Not all the best answers in cQA are really best answers.

Another popular method adopted by most cQA service is list all the similar questions. In this way users could browse all the questions and find useful information to their questions. However, this method requires users to browse several question pages, which costs users much time and thus is not suitable for those who are of urgent information need, e.g., mobile internet users. Users, particularly mobile internet users, prefer direct, succinct and comprehensive answers to a ranked list question. Yang Tang (Tang et al., 2010) proposed that summarizing answers with similar questions is of great importance. And summarizing answers from those similar questions are supposed to be a good method to solve the problems listed above. Although the questions may be not exact the same, or their answers are not best, their answers are still useful to user's question to some extent.

The remainder of the paper is organized as follows. In Section 2 we provide a brief overview of the whole system and

[1]CLEF (2009). http://www.clef-campaign.org. Accessed 2009.

[2]TREC (2009). http://trec.nist.gov/. Accessed 2009.

[3]NTCIR (2009). http://research.nii.ac.jp/ntcir/. Accessed 2009.

we introduce the data set used in the system. Section 3 presents the methods to analyze questions and retrieve similar questions. The summarization method will be described in detail in Section 4. Finally in section 5 we will make a conclusion.

2. System Overview

In this paper, we develop an answer retrieval and summarization system which directly provides an accurate and comprehensive answer summary over a large Chinese cQA database.

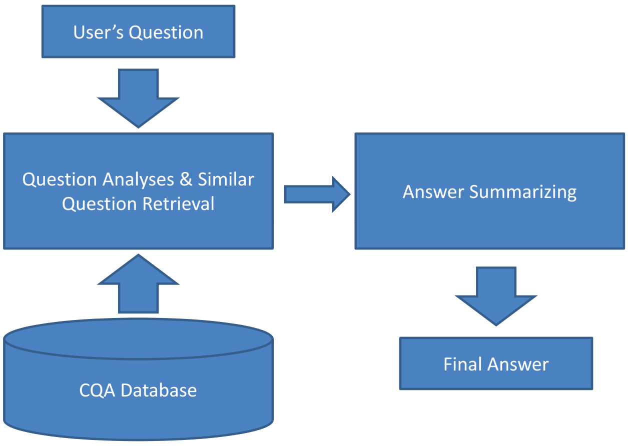

Figure 1: System framework

As figure1 shows, when a user's question input, the system will work as following workflow. First, the system will analyze the question, and with the analysis result it will retrieve a list of similar questions from a large cQA database. Then, according to different question and the level of similarity of questions, the system will adopt different strategies to make summarizations from answers of the similar questions. Finally the system will output a brief and accurate answer summary.

In order to implement such a system, we use over 10 million questions and their answers from different cQA sites. To store these questions and answers, we distributed them on several machines.

3. Similar Question Retrieval

To get potential answers to a given question, we first retrieve similar questions from local data. This is quite different from traditional QA systems. Our question retrieval module workflow is shown in Fig. 2.

First we analyze the given question and extract keywords from it. Then we formulate queries from the keywords and search candidate questions in local data. Next, we classify the given question and each candidate into predefined question types. Finally, we rerank the candidates by lexical similarities and question types. The top-n candidates after reranking are returned as similar questions.

3.1. Question Analysis

In the case of Chinese, we first do word segmentation and POStagging on the given question. We use the Chinese segmentation and POStagging tool developed by Institute of Computing Technology Chinese Academy of Sciences which can be downloaded via its homepage[4] .

From the tagged question, we extract "important" words which includes nouns, verbs, long(e.g. more than one character) adjectives, numbers and times. All "important" words are sorted by their TF-IDF and at most three words on the top are collected as keywords. The TF-IDF values are precalculated.

3.2. Question Retrieval

In Question Retrieval stage, we formulate two queries from a set of keywords by two strategies. For first query, we do synonym expansion for every keywords. Chinese synonyms are gotten from Cilin dictionary API provided by Harbin Institute of Technology[5]. We search for the questions in which every keywords (or its synonym) must appears. For second, we just to find the questions in which the keywords appears as much as possible.

For example, given a question 贝克汉姆的妻子是谁 "Who is Beckham's wife", the keywords are 贝克汉姆 "Beckham" and 妻子 "wife". From Cilin dictionary we know that 贝克汉姆 "Beckham" has no synonym and the synonym of 妻子 "wife" is 老婆 "wife(informal)". Express them by the syntax of Lucene, the two queries are: 1. "+"贝克汉姆" +("妻子" OR "老婆")" ; 2 ""贝克汉姆" "妻子"".

Although the first query is under synonym expansion, it is more rigid than the second query in practice. So we first collect candidates from the results of first query and then the other. The size of candidate set is no larger than 100.

3.3. Question Classification

Before reranking, we classify the original question and each candidate question into predefined question types. Traditional question categorization system is relate to content, such as the UIUC taxonomy(Li and Roth, 2005). In our system, the question are classified according to the user's purpose. That is to say, what we concerned is what the users are asking and which types of answer summaries would satisfy them, our taxonomy contains five types as follows:

[4] http://www.ictclas.org/
[5] http://ir.hit.edu.cn/

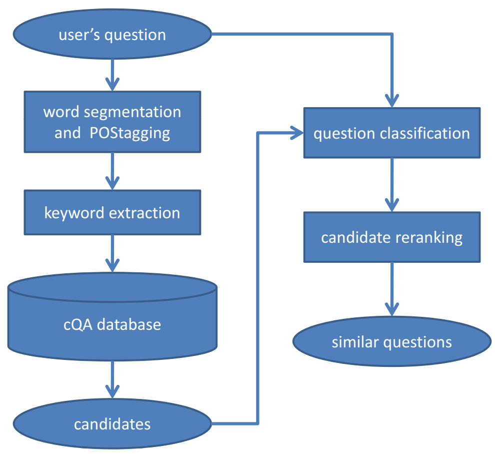

Figure 2: Workflow of Similar Question Retrieval

1. for Fact

   People ask this kind of question just for general facts, such as “Who is the president of United States” or “List the names of Nobel price winners in 1990s”. The expected answer would be a short phrase or a list of short phrases.

2. for Reason

   Sometimes people ask questions not for answers but opinions or explanations. So only returning the answer is not satisfactory. For example, when asking “Is it good to drink milk while fasting?”, the questioner prefers to “It is not a good idea due to the fact that lactase is a common food allergen which may cause some problems...” rather than just a word “No”. Some questions even have no correct answers, such as “Who will be the NBA MVP of this season?”. A good answer summary for this question should contain all kinds of opinions instead of a most probable answer.

3. for Solution

   Sometimes people encounter troubles in their life and want to find solutions on the web. The answer to these questions usually holds a inner older (e.g. ”To solve the problem, you should first..., second...”). Thus, the summary task should be choosing a typical answer instead of selecting reasonable sentences from answers.

4. for Definition

   People ask this kind of questions for the definition of a concept or the different between concepts. Generally, We don’t need to summarize answers for definition question. However, if the best answer is too long to display, it is better to be summarized into a short one.

5. for Resource

   People ask this kind of questions for resources instead of information (e.g. ”Where can I download the StarCraft 2 demo?”). The typical answer is navigational (name or url of a website). These answers can be extracted by regular expressions.

Temporarily, we manually build templates for each question types. A question is classified to a type if it matches one of the corresponding templates. Otherwise, it is tagged as UNCLASSIFIED. Currently, our templates approximately covers 65 percent of questions.
Question classification is indispensable in our system. If the type of a candidate is different from the given question, its answers are unlikely to be acceptable. What’s more, the question type is a useful information for answer summarization. We can employ different summary strategies on different type.

3.4. Candidate Reranking

After candidate retrieval and question classification, we rerank all candidates by their lexical similarity with the original question. We use four similarity metrics listed as follows:

- Cosine Similarity (CS)

  Cosine Similarity is a measure of similarity between two vectors by finding the cosine of the angle between them. It is often used to compare documents in text mining. Given two vectors of attributes A and B, the Cosine Similarity CS is defined as

  $$\mathrm{CS}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}.$$

- Maximum Overlap (MO)

  Maximum Overlap is also a similarity measure between two vectors. Given two vectors of attributes A and B, the Maximum Overlap MO is defined as

  $$\mathrm{MO}(A, B) = \frac{\sum_{m \in A, m \in B} (A(m) + B(m))}{\sum_{a \in A} A(a) + \sum_{b \in B} B(b)},$$

  where $A(a)$ stands for the the value of attribute a in vector A.

- Longest Common Substring Similarity(LCSsim)

  LCSsim measures the similarity between two strings, which is different from CS and MO. Given two strings s_1 and s_2, the LCSsim is defined as

  $$\mathrm{LCSsim}(s_1, s_2) = \frac{\mathrm{length}(\mathrm{LCS}(s_1, s_2))}{\max(\mathrm{length}(s_1), \mathrm{length}(s_2))},$$

  where $\mathrm{LCS}(s_1, s_2)$ is the longest common substring of s_1 and s_2.

- Longest Consecutive Common Substring Similarity(LCCSsim)

  Given two strings s_1 and s_2, the $\mathrm{LCCSsim}$ is defined as

  $$\mathrm{LCCSsim}(s_1, s_2) = \frac{\mathrm{length}(\mathrm{LCCS}(s_1, s_2))}{\max(\mathrm{length}(s_1), \mathrm{length}(s_2))},$$

  where $\mathrm{LCCS}(s_1, s_2)$ is the longest consecutive common substring of s_1 and s_2.

For each question, we do word segmentation and remove punctuations. All words are collected as word vector and all important words are collected as (mentioned in Section 3.1.) important word vector. The weight of each word is its length. Long words are more important than short words. We measure six similarities between original question and each candidates: CS between word vectors, MO between word vectors, LCSsim between questions, LCCSsim between questions, CS between important word vectors and MO between important word vectors. The overall similarity is the linear combination of this six measures. In practice, we find similarity between word vectors is more reliable than the others. So we heuristically set the weight of first two measures to 0.25 and the rest to 0.125. Obviously, the overall similarity is between 0 and 1. If the original question and a candidate are classified into different types, the similarity of the candidate is divided by 2.

4. Summarization Subsystem

Give a question (a user's query) and a set of its similar questions and their answers, the task of summarization subsystem is to generate the answer to this question. Two different strategies are taken according to the classification result during the query analyzing process: 1) if the first similar question is close to the query question, the best answer (ranked first) of the first similar question will be returned; 2) if the quality of the first similar question is not good enough, or the type of the question needs to extract information from more than one question, the general summarization framework will be used to extract sentences to form a good answer. Our general summarization framework is based on the theory of Kolmogorov complexity and information distance. Next we will introduce our information distance based summarization framework.

4.1. Theory

Fix a universal Turing machine U. The Kolmogorov complexity (Li and Vitányi, 1997) of a binary string x conditioned to another binary string y, $K_U(x|y)$, is the length of the shortest (prefix-free) program for U that outputs x with input y. It can be shown that for a different universal Turing machine U', for all x, y

$$K_U(x|y) = K_{U'}(x|y) + C,$$

where the constant C depends only on U'. Thus $K_U(x|y)$ can be simply written as $K(x|y)$. We write $K(x|\epsilon)$, where ϵ is the empty string, as $K(x)$. It has also been defined in (Bennett et al., 1998) that the energy to convert between x and y to be the smallest number of bits needed to convert x to y and vice versa. That is, with respect to a universal Turing machine U, the cost of conversion between x and y is:

$$E(x,y) = \min\{|p| : U(x,p) = y, \; U(y,p) = x\} \quad (1)$$

The following theorem has been proved in (Bennett et al., 1998):

Theorem 1 $E(x,y) = \max\{K(x|y), K(y|x)\}$.

Thus, the max distance was defined in (Bennett et al., 1998):

$$D_{\max}(x,y) = \max\{K(x|y), K(y|x)\}. \quad (2)$$

4.2. Summarization Framework

Our framework is based on our newly developed theory of information distance among many objects. In this section we will firstly introduce our newly developed theory and then our summarization model based on the new theory.

4.2.1. Theory

In (Long et al., 2008), the authors generalize the theory of information distance to more than two objects. Similar to Equation 1, given strings $x_1, \ldots, x_n$, they define the minimal amount of thermodynamic energy needed to convert any x_i to any x_j as:

$$E_m(x_1, \ldots, x_n) = \min\{|p| : U(x_i, p, j) = x_j \text{ for all } i, j\} \quad (3)$$

Then it is proved in (Long et al., 2008) that:

Theorem 2 *Modulo to an $O(\log n)$ additive factor,*

$$\min_i K(x_1 \ldots x_n | x_i) \le E_m(x_1, \ldots, x_n) \le \min_i \sum_{k \neq i} D_{\max}(x_i, x_k) \quad (4)$$

Given n objects, the left-hand side of Equation 4 may be interpreted as the most comprehensive object that contains the most information about all of the others. The right-hand side of the equation may be interpreted as the most typical object that is similar to all of the others.

4.2.2. Modeling

We have developed the theory of information distance among many objects. In this subsection, a new summarization model be built based on our new theory.

The task of summarization can be described as follows: given n documents $B = \{B_1, B_2, \ldots, B_n\}$, the task requires the system to generate a summary S of B. According to our theory, the conditional information distance among $B_1, B_2, \ldots, B_n$ is $E_m(B)$.

However, it is very difficult to compute E_m. Moreover, E_m itself does not tell us how to generate a summary. Equation 4 has provided us a feasible way to approximate E_m: the most comprehensive object and the most typical one are the left and right of Equation **??**, respectively. The most comprehensive object is long enough to cover as much information in B as possible, while the most typical object is a concise one that expresses the most common idea shared by those objects. Since we aim to produce a short summary to represent the general information, the right-hand side of Equation 4 should be used. The most typical document is the B_j such that

$$\min_j \sum_{i \neq j} D_{\max}(B_i, B_j)$$

However, B_j is far from enough to be a good summary. A good method should be able to select the information from B_1 to B_n to form a best S. We view this S as a document in this set. Since S is a short summary, it does not contain extra information outside B. The best traditional summary S_{trad} should satisfy the constraint as:

$$S_{trad} = \arg\min_S \sum_i D_{\max}(B_i, S) \quad (5)$$

In most applications, the length of S is confined by $|S| \le \theta$ (θ is a constant integer) or $|S| \le \alpha \sum_i |B_i|$ (α is a constant real number between 0 and 1).

We have already developed a framework for summarization. However, the problem is that neither $K(.)$ nor $D_{max}(.,.)$ is computable. we can use frequency count, and use Shannon-Fano code (Cilibrasi and Vitányi, 2007) to encode a phrase which occurs in probability p in approximately $-\log p$ bits to obtain a short description.

4.3. Computing Information Distance

we can use frequency count, and use Shannon-Fano code (Page 67, Example 1.11.2 in (Li and Vitányi, 1997)) to encode a phrase which occurs in probability p in approximately $-\log p$ bits to obtain a short description. Firstly we divide a sentence into semantic elements; then information distance between two sentences is estimated through their semantic element sets.

4.3.1. Semantic Element Extraction

In a document, each word or entity contains a certain amount of information, and the information varies according to the word or entity's importance. Such words or entities are called "semantic elements", and "elements" for short in this paper.

We use Stanford Named Entity Recognition (NER)[6] to extract semantic elements from English documents, and word segmentation method to extract words from Chinese documents.

4.3.2. Information Distance Approximation

Next we will take several steps to do the approximations. Although some steps contain rough approximations, we will investigate the influence of our estimations with extensive experiments in Section **??**.

Let $M = \{M_1, M_2, \ldots\}$ and $N = \{N_1, N_2, \ldots\}$ to be two sets of sentences. After those steps mentioned in Section 4.3.1., each sentence M_i (or N_j) has an element set M_i^* (or N_j^*). According to Equation 2,

$$D_{\max}(M, N) = \max\{K(M|N), K(N|M)\},$$

then

$$\begin{array}{l} K(M|N) \approx K(\bigcup_i M_i^* \setminus \bigcup_j N_j^*), \\ K(N|M) \approx K(\bigcup_j N_j^* \setminus \bigcup_i M_i^*). \end{array} \tag{6}$$

The Kolmogorov complexity of an element set W can be computed by the sum of the complexities of all its elements:

$$K(W) = \sum_{w \in W} K(w)$$

According to the the coding theory, the complexity of an element w can be computed by its probability (Li and Vitányi, 1997), which can usually be approximated by its document frequency in the corpus:

$$K(w) = -\log P(w) \approx -\log df(w) \tag{7}$$

5. Conclusion

In this paper, we presented, to the best of our knowledge, the first QA system which summarize answers with similar questions. The contribution of this paper is three-fold: (a) We proposed that cQA can serve as a good resources to solve some complicated questions, its questions can be treated as important users' query log. (b) we introduce a new method to reuse existing question. Instead of retrieving similar questions, we believe that answer summarization may be a better approach to reuse existing question and answer resources. (c) We design a system which summarize answers from similar questions according to different situations.

As the future work, we will incorporate more information and relations into our question retrieval and summarization model. Furthermore, we plan to utilize more sophisticated summarization approaches to improve the quality of summaries.

Acknowledgments

This work was partly supported by the Chinese Natural Science Foundation under grant No.60973104 and partly carried out with the aid of a grant from the International Development Research Center, Ottawa, Canada IRCI project from the International Development.

6. References

Charles H. Bennett, Péter Gács, Ming Li, Paul M.B. Vitányi, and Wojciech H. Zurek. 1998. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423, July.

Rudi L. Cilibrasi and Paul M.B. Vitányi. 2007. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, March.

Xin Li and Dan Roth. 2005. Learning question classifiers: the role of semantic information. *Natural Language Engineering*.

Ming Li and Paul M.B. Vitányi. 1997. *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag.

Chong Long, Xiaoyan Zhu, Ming Li, and Bin Ma. 2008. Information shared by many objects. In *CIKM*, pages 1213–1220.

Yang Tang, Minlie Huang, and Xiaoyan Zhu. 2010. Summarizing answers with similar questions for chinese cqa portals. unpublished, has submitted to ITCS2010.

[6]http://nlp.stanford.edu/ner/index.shtml

Yahoo! Answers for Sentence Retrieval in Question Answering

Saeedeh Momtazi, Dietrich Klakow

Spoken Language Systems
Saarland University, Saarbrücken, Germany
{saeedeh.momtazi,dietrich.klakow}@lsv.uni-saarland.de

Abstract

Question answering systems which automatically search for user's information need are considered as a separate issue from the community-generated question answering which answers users' questions by human respondents. Although the two answering systems have different applications, both of them aim to present a correct answer to the users' question and consequently they can feed each other to improve their performance and efficiency. In this paper, we propose a new idea to use the information derived from a community question answering forum in an automatic question answering system. To this end, two different frameworks, namely the class-based model and the trained trigger model, have been used in a language model-based sentence retrieval system. Both models try to capture word relationships from the question-answer sentence pair of a community forum. Using a standard TREC question answering dataset, we evaluate our proposed models on the subtask of sentence retrieval, while training the models on the *Yahoo! Answer* corpus. Results show both methods that trained on Yahoo! Answers logs significantly outperform the unigram model, in which the class-based model achieved 4.72% relative improvement in mean average precision and the trained triggering model achieved 18.10% relative improvement in the same evaluation metric. Combination of both proposed models also improved the system mean average precision 19.29%.

1. Introduction

While information retrieval historically focuses on searching for relevant documents, it is often the case that only a portion of a *relevant* document is related to a user's information need. In such a situation, it may be preferable instead to retrieve only the relevant portion of the document which includes the information that the user requires. Such an idea has recently motivated researchers to develop question answering systems which retrieve the exact information required by the users.

Within a question answering system, document retrieval is an important component which should provide a list of candidate documents to be analyzed by the rest of the system. Document retrieval, however, is insufficient, as the retrieved documents are much larger than the required answer, and topic changes typically occur within a single document. In addition, in the question answering context, the relevant information is most often found in one sentence or two. Hence, it is essential to split the text into smaller segments, such as sentences, and rank them in a sentence retrieval step. This is the focus of our research. Retrieved sentences are then further processed using a variety of techniques to extract the final answers. It has been proven that this component has an important role in a question answering system such that improvement in sentence retrieval performance has a significant positive effect on the accuracy of the question answering system (Shen, 2008). Figure 1 shows a simple structure of a question answering system considering both levels of the information retrieval component and also the information extraction component.

Although available retrieval methods used for document retrieval are also applicable for the task of sentence retrieval, the performance of these models in the sentence retrieval task is worse than for the task of retrieving documents. Because there are major differences between document retrieval and sentence retrieval which affect their performance. As a result, many of the assumptions made about

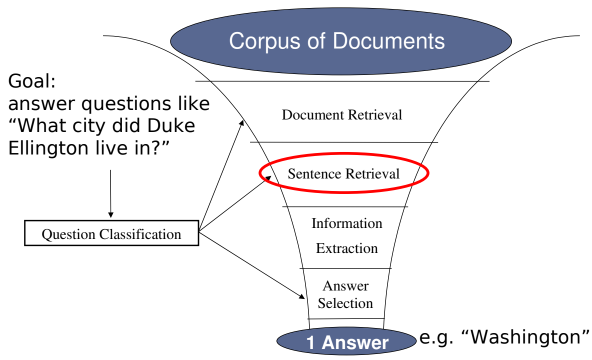

Figure 1: The Structure of a Question Answering System.

document retrieval do not hold for sentence retrieval (Murdock, 2006). The brevity of sentences vs. documents is the most important feature that exacerbates term-mismatch problems.

In the common retrieval methods, a search is performed for only the exact literal words presented in the query. Consequently, these algorithms fail to retrieve other relevant information. For example, consider the sample question in Table 1 and its correct answer sentences. The sentence retrieval component might retrieve the first sentence, because there are two shared terms, *"invented"* and *"car"*, between the question and this sentence. Using the exact matching models, however, the retrieval algorithm misses the second and the third sentences, because these sentences do not share any term with the question. In other words, although these sentences contain the words *"built"*, *"vehicle"*, and *"automobile"* which are very likely to be relevant to the question terms, the sentence retrieval model is not able to recognize their relationship.

While different approaches such as automatic query expansion have been a great success story for solving the term-

Table 1: A sample question and possible answer sentences in a search space

Question	*"Who invented the car?"*
Answer	*"Between 1832 and 1839, Robert Anderson of Scotland invented the first crude electric car carriage."*
Answer	*"Nicolas-Joseph Cugnot built the first self propelled mechanical vehicle."*
Answer	*"An automobile powered by his own engine was built by Karl Benz in 1885 and granted a patent."*

mismatch problem in document retrieval, attempts to apply them to sentence retrieval have had rather mixed success (Murdock, 2006). For this reason, it is desirable to have a more sophisticated model to capture the semantics of sentences rather than just the term distributions. This issue has motivated a great deal of research on term relationships over the last decades. However, improvements in system performance from such schemas have proven challenging, for two primary reasons: the difficulty of estimating term relationships and the difficulty of integrating both exact match and term relationships in a single weighting schema (Gao et al., 2004).

Various research has been done on estimating of term relationships for information retrieval, as will be described in Section 2.2. In the task of question answering, however, it is more difficult to find relevant sentences. This is due to the fact that there are so many sentences in the search space that are relevant to the question, but do not include the answer. In this case, it is necessary to find a novel information resource which is closer to the question answering purpose. We believe that although community-generated question answering and automatic question answering systems use two separate approaches, the community question answering forums that collect answers from human respondents are a useful resource that can be exploited to the term-mismatch problem of the automatic systems. Generally, community-generated question answering forums provide very informative corpora that contain pairs of question-answer sentences. As the question and its relevant answer sentences typically talk about the same topic, there is a latent relation between the question words and the terms appearing in the answer sentences, even though there are not many common terms between the pair of the question and the answer sentences. In this paper, we propose a novel approach to use community question answering logs in two different language model-based frameworks, namely class-based model and trained triggering model, and apply them to the sentence retrieval task.

The paper is organized as follows. In Section 2 we review related work using both the general language modeling approaches for information retrieval and term relationship approaches. Section 3 and 4 describe the class-based and trained triggering models that we use to capture word relation from community question answering logs. The dataset, corpus, and experimental results are presented in Section 5. Finally, Section 6 summarizes the paper.

2. Related Work

2.1. Language Models for Information Retrieval

Statistical language modeling has successfully been used in speech recognition (Jelinek, 1998) and many natural language processing tasks including part of speech tagging, syntactic parsing (Charniak, 1993), and machine translation (Brown et al., 1990).

Language model for information retrieval has received researchers' attention during the recent years. The efficiency of this approach, its simplicity, the state-of-the-art performance, and clear probabilistic meaning are the most important factors which contribute to its popularity (Lafferty and Zhai, 2001; Ponte and Croft, 1998).

The idea of using language model techniques for information retrieval applications was proposed by Ponte and Croft (1998). They inferred a language model for each document and estimated the probability of generating the query according to each of these models. In their method, each query is considered as a set of unique terms and two different probabilities are computed. The first one is the probability of producing the query terms; and the second one is the probability of not producing other terms. Then they use the product of these two factors as their model. In addition, Hiemstra (1998) considered each query as a sequence of terms and computed the query probability by multiplying the probability of each individual term. Song and Croft (1999), and Miller (1999) also used the same method.

Berger and Lafferty (1999) proposed a translation-based approach which computes the probability of generating a query as a translation of a document. They utilized this probability as a measure of relevance of a document to a query to rank the documents. Following this method, Lafferty and Zhai (2001) proposed another technique to extend their current model by estimating the language models of both documents and queries. In this approach, the language models of documents and queries are computed and then the Kullback-Leibler divergence between the probabilities of document model and query model is used.

Zhai and Lafferty (2001) estimated the conditional probability $P(D|Q)$ by applying the Bayes' formula and dropping a document-independent constant:

$$P(D|Q) \propto P(Q|D)P(D) \tag{1}$$

where $P(Q|D)$ is the probability of the query given a document and $P(D)$ is the prior probability of a document. Since $P(D)$ is assumed to be uniform for ranking the documents, it will be ignored in further computations.

$P(Q|D)$ is the probability of generating a query Q given the observation of the document D; and the documents are ranked in descending order of this probability. The word-based unigram model estimates the probability of generating the query by:

$$P(Q|D) = \prod_{i=1...M} P(q_i|D) \tag{2}$$

where M is the number of terms in the query, q_i denotes the ith term of query Q, and D is the document model (Song and Croft, 1999). Merkel and Klakow (2007) used the same model for the sentence retrieval task. Since in this case the documents are divided into sentences, $P(Q|S)$ is computed; where S is the sentence to be ranked. The unigram model for sentence retrieval computes the probability of the query Q given the sentence S by:

$$P(Q|S) = \prod_{i=1...M} P(q_i|S) \tag{3}$$

2.2. Term Relationship Models for Information Retrieval

As mentioned in the previous section, by estimating the word-based unigram model, the ranking algorithms only try to match the literal words that are present in queries and texts; but they will fail to retrieve other relevant information. To avoid this problem, researchers have tried to apply methods using different types of term relationships to the retrieval task.

Using hand-crafted thesauri such as WordNet is one of the prevalent techniques (Mandala et al., 1998; Schütze and Pedersen, 1997). The thesaurus is incorporated in different retrieval models to find the dependencies among the words. Robertson et al. (1981) used a thesaurus for the probabilistic retrieval model; Cao et al. (2005; 2006), and Croft and Wei (2005) applied it for the language model-based retrieval. Voorhees (1994) used WordNet for query expansion; and Liu et al. (2004) used this approach to disambiguate word senses by expanding the query. Since the results of using manual processing provide useful information, it seems like a good method for our goal. However, manual processing causes many problems such as inconsistency and ambiguity (Jones, 1971). The absence of proper nouns causes another problem; since WordNet and most of the thesauri do not consider proper nouns and we cannot find relationship between these nouns and other terms as a result. In addition, we can not find the measure of dependency between the terms in the manual processing, since they only give us a binary classification of relevant and non-relevant terms to an special term. Beside these problems, building a thesaurus is labor intensive. Because of such problems, an automatic method is more desirable.

Grammatical analysis has also been applied as an automatic approach to find dependencies between terms in a sentence. Nallapati and Allan (2002) introduced a new probabilistic sentence tree language model approach. Gao et al. (2004) described a linkage model to relax the independence assumption. Although grammatical analyses can provide very specific knowledge about term relations, they are not robust (Manning et al., 2008) and also need a deep sentence processing.

The use of co-occurrence statistics is another well-known method which focuses on term relations. Cao et al. (2005) used the co-occurrence relations and integrated them with the relations extracted from WordNet. Wei and Croft (2007) introduced a probabilistic term association measure and utilized this measure in document language models. Van Rejsbergen. (1979), and Burgess et al. (1998) also used words co-occurrence in window scaling. Qui and Frei (1993) applied another similar method to expand a query. In their proposed method, each new query term takes the same weight as its similarity to the original query term. Chung and Chen (2002) described another technique called correlation-verification smoothing to find correlations among terms. Since the term co-occurrence method is a window-based approach, finding a suitable window size automatically is not easy (Wei and Croft, 2007).

For applying term relations, some researchers also tried to use document reformulation. Cluster-based document models (Liu and Croft, 2004; Tao et al., 2006) and LDA-based document models (Wei and Croft, 2006) are two important models in this area. They are both expensive, especially for large collections.

Momtazi and Klakow (2009) proposed the class-based language model by applying term clustering. This model is found to be effective in capturing relevant terms. The flexibility of this model in using different types of word co-occurrence (Momtazi et al., 2010) offers a distinct advantage as it is also adaptable for question-answer pair co-occurrence which is our goal.

Trained trigger language model is another approach recently proposed for sentence retrieval and proven to outperform the unigram model. As this model can also be trained on a question-answer pair corpus, it is a useful framework for our task. In the next sections we will describe both class-based and trained trigger models in more detail.

3. Class-based Model

The idea of class-based model is clustering similar words together to reduce the term-mismatch problem. Partitioning vocabulary into a set of word clusters, the sentence retrieval engine can retrieve sentences which do not contain question words, but their terms are in the same clusters as question words.

As mentioned, in the basic language model-based sentence retrieval, the word-based model, $P(Q|S)$ is estimated based on the probability of generating each query term q_i conditioned on a candidate sentence S. In class-based unigrams, $P(Q|S)$ is computed using only the *cluster labels* of the query terms as

$$P(Q|S) = \prod_{i=1...M} P(q_i|C_{q_i}, S)P(C_{q_i}|S), \tag{4}$$

where C_{q_i} is the cluster containing q_i and $P(q_i|C_{q_i}, S)$ is the emission probability of the ith query term given its cluster and the sentence. $P(C_{q_i}|S)$ is analogous to the sentence model $P(q_i|S)$ in (3); however in this model, the probability is calculated based on clusters instead of terms. To calculate $P(C_{q_i}|S)$, each cluster is considered an atomic entity, with Q and S interpreted as sequences of these entities (Momtazi and Klakow, 2009).

To cluster lexical items, we use the algorithm proposed by Brown et al (1992), as implemented in the SRILM toolkit (Stolcke, 2002). The Brown algorithm uses mutual information between cluster pairs in a bottom-up approach to maximize *Average Mutual Information* between adjacent clusters. Algorithm 1 shows the details of the Brown clustering.

The algorithm requires an input corpus statistics in the form $\langle w, w', f_{ww'} \rangle$, where $f_{ww'}$ is the number of times the word w' is seen in the context w. Both w and w' are assumed to come from a common vocabulary.

Algorithm 1 The Brown Word Clustering Algorithm
(AMI stands for Average Mutual Information)

Initial Mapping: Put a single word in each cluster
Compute the initial AMI of the collection
repeat
  Merge a pair of clusters which has the minimum decrement of AMI
  Compute AMI of the new collection
until reach the predefined number of clusters K
repeat
  Move each term to the cluster for which the resulting partition has the greatest AMI
until no more increment in AMI

As shown in Algorithm 1, the clusters are initialized with a single term. Then, a bottom-up approach is used to merge the pair of clusters that minimizes the loss in average mutual information between the word cluster $C_{w'}$ and its context cluster C_w. Different words seen in the same contexts are good candidates for the merger, as there are different contexts in which the same words are seen. This step continues for $V - K$ iterations, where V is the number of terms and K is the predefined number of clusters. To increase the average mutual information, a final step is performed, whereby each term is moved to that cluster for which the resulting partition has the greatest average mutual information. The algorithm terminates when average mutual information ceases to increase.

While originally proposed with bigram statistics, the algorithm is *agnostic* to the definition of co-occurrence and several notions of co-occurrence can be used to cluster words (Momtazi et al., 2010). For example if $\langle w, w' \rangle$ are adjacent words, the algorithm clusters words based on their surroundings terms; if $f_{ww'}$ is the number of times w and w' appear in the same document, it will produce semantically (or topically) related word-clusters. Since we want to apply this class-based model to the sentence retrieval for question answering system, the pair of question and answer sentences is an informative resource for this task. In this model, the questions' terms that have the same words in their answer sentence are clustered together and also the answers' terms that have the same words in their related question are clustered together.

Considering the community question answering forums as a set of question-answer sentence pair, we can say w and w' co-occurred if the word w appears in the question and w' appears in the related answer. Because if the two content words w and w' are seen in the pair of question and answer sentence, they are usually topically related.

Statistics of this co-occurrence may be collected in two different ways. In the first case, $f_{ww'}$ is simply the number of question-answer pairs that contain both w and w'. Alternatively, we may want to treat each *instance* of w' in an answer sentence that contains an instance of w in its question to be a co-occurrence event. Therefore, if w' appears three times in an answer sentence that contains two instances of w in its question, the former method counts it as one co-occurrence, while the latter as six co-occurrences.

We use the latter statistic, since we are concerned with retrieving sentence sized information, wherein a repeated word is more significant.

4. Trained Trigger Model

The goal of the trained trigger model is using wider information to relax the exact matching assumption. The available question-answer sentence pairs is one of the most informative resource that can be used for finding pairs of trigger and target words. In this model, each word in the question triggers all of the answer words.

Such a model can retrieve sentences which have no or a few words in common with the question but their terms have frequently co-occurred with question terms in the pairs of question-answer sentences used for training the model.

As an example, consider the following question and its correct answer sentence.

Q: *How high is Everest?*
A: *Everest is 29,029 feet.*

We see the above question and the answer sentence share a very limited number of terms, the term *"Everest"* in this example. In such a situation, it is very unlikely that the basic query likelihood model ranks the correct answer on the top of the list. Because there are a lot of irrelevant sentences in the search space which contain the same word such as:

S: *Everest is located in Nepal.*
S: *Everest has two main climbing routes.*

However, the triggering model which is trained on a question-answer sentence pair corpus gives a higher score to the correct sentence because the model knows that in a large portion of questions that contain the word *"high"*, the term *"feet"* appear in the answer. As a result, in the trained model, the word *"high"* triggers the target word *"feet"*. The following sentences are some of the samples that can be found in a training corpus:

Q1: *How high is Mount Hood?*
A1: *Mount Hood is in the Cascade Mountain range and is 11,245 feet.*

Q2: *How high is Pikes peak?*
A2: *Pikes peak, Colorado At 14,110 feet, altitude sickness is a consideration when driving up this mountain.*

In the basic language model-based sentence retrieval, for each sentence in the search space a language model is trained, and then using the maximum likelihood estimation, $P(q_i|S)$ is calculated based on the frequency of query term q_i in sentence S:

$$P(q_i|S) = \frac{c(q_i, S)}{\sum_w c(w, S)} \tag{5}$$

where $c(q_i, S)$ is the frequency of ith query term in sentence S.

In trained trigger language model, contrary to the maximum likelihood, first a single model is trained on a large corpus, then it is being used for all of the sentences to be retrieved. The trained model is represented by a set of triples $< w, w', f_{ww'} >$, where $f_{ww'}$ is the number of times the word w triggers the target word w'. Having such a trained model, the language model-based sentence retrieval is reduced to:

$$P(q_i|S) = \frac{1}{N} \sum_{j=1...N} P_{trigger}(q_i|s_j) \quad (6)$$

where s_j is the jth term in the sentence and N is the sentence length. In this model $P(q_i|s_j)$ is calculated as follows:

$$P_{trigger}(q_i|s_j) = \frac{1}{M} f_{q_i,s_j} \quad (7)$$

where M is the query length and f_{q_i,s_j} is the number of times the query term q_i triggers the sentence word s_j based on the training corpus. As in the maximum likelihood model, any smoothing method can be used. In all of our experiments we use Bayesian smoothing with Dirichlet prior (Lafferty and Zhai, 2001).

5. Experimental Results

5.1. TREC Question Answering Dataset

To evaluate our model, we used the set of questions from the TREC[1] 2006 question answering track as the test data, while the TREC 2005 set was used for development. The TREC 2006 question answering task contains 75 question-series, each on one topic, for a total of 403 factoid questions. These questions were used as queries for sentence retrieval.

Since the relevance judgments released by NIST are only at the document level (Dang et al., 2006), we required another annotated corpus for sentence-level relevance judgments. To this aim, the *Question Answer Sentence Pair* corpus of Kaisser and Lowe (2008) was used. All the documents that contain relevant sentences are from the NIST AQUAINT1 corpus.

Question answering systems typically employ sentence retrieval after initial, high quality document retrieval. To simulate this, we created a separate *search collection* for each question using all sentences from all documents that are relevant to the topic (question-series) from which the question was derived. On average, there are 17 documents / 270 sentences that are relevant to each question topic (documents which are relevant to any of 5-6 different questions in a question-series) while the number of relevant sentences to each individual question is only 4 sentences (on average). So that for each question there are several irrelevant documents: they may be relevant to another question. Furthermore, irrelevant documents to a question are relevant to a related question, and hence are typical of the false alarms that would arise if one were retrieving documents based on one of the questions. As a result, the sentence search collection is realistic, even if somewhat optimistic.

[1]See *http://trec.nist.gov*.

5.2. Corpus for Clustering and Triggering

To cluster lexical words for the class-based model and also train the triggering language model, we used *Yahoo! Answers Comprehensive Questions and Answers corpus [version 1.0]*. This dataset derived from *http://answers.yahoo.com/* which is a web site where people post questions and other people can answer their questions. This web site is public to any web user willing to browse or download them (Webscope, 2009).

The Yahoo! Answers corpus has been collected in 10/25/2007. It includes all the questions and their corresponding answers. The size of the corpus is 3.45 GB while containing 4,483,032 questions and their answers. In addition to the question and the answer texts, the corpus contains a small amount of meta data, i.e., which answer was selected as the best answer, and the category and the subcategory assigned to each question. No personal information is included in the corpus. Table 2 shows a sample content in the Yahoo! Answers corpus, in addition to the meta information. In our experiments, both subject and content are considered as question and all answers including the best one are used as answer sentences.

Such a dataset has been used for different task like learning and validating answer extraction models (Surdeanu et al., 2008). However, to the best knowledge of the author this dataset has not been used for the sentence retrieval in question answering systems.

For both class-based and triggering model, we used this corpus to extract the pair of words in which the first word is a question term and the second word is the answer sentence term in addition to their frequency. In the class-based model, the extracted word pairs are used as the input of Brown clustering and then the clusters are used for retrieving more relevant sentences. In the triggering model, however, the word pairs are directly used in the sentence retrieval model, in which if the first word (trigger word) appears in the question and the second word (target word) appears in a sentence, we consider that there is a relation between the question and the sentence.

5.3. Building the Model

We evaluated our proposed models in two scenarios: on their own and in combination with the word-based model in which each of the models were interpolated with the original word-based model (Equation 3). We believe that even though using the information derived from community question answering logs can improve the sentence retrieval performance, it is necessary to keep the regular exact matching model too. Because this original model can give priority to the words that are shared between the question and the answer sentence, as they still play an important role in the system.

In this experiment, the word-based unigram model with maximum likelihood estimation is considered as the baseline. For the class-based model, the class-based unigram with maximum likelihood estimation is calculated and for the trained triggering model, the word-based unigram with triggering estimation is used. For all of the experiments Bayesian smoothing with Dirichlet prior (Mackay and Peto, 1995) has been applied and then linear interpolation (Je-

Table 2: A sample content from Yahoo! Answers

Subject	*"What are world's 3 smallest nations?"*
Content	*"i.e. which is the smallest nation? which is the next one in size? and then the next one? How did they manage to become countries?"*
Best Answer	*"1. The Vatican City in central Rome ranks as the smallest nation of the world. It measures 0.17 square miles. 2. Monaco. 3. San Marino. How did they manage to become countries? 1. The political freedom of the Vatican is guaranteed and protected by Italy. It concordat between the Holy See and the kingdom of Italy signed in 1929 in the Lateran Palace, Rome, by Cardinal Gasparri for Pius XI and by Benito Mussolini for Victor Emmanuel III. 2. Monaco came under French protection in 1861. However from 1911, when the first constitution was promulgated, the prince of Monaco was an absolute ruler. 3. According to tradition, Marino, a Christian stonecutter from Dalmatia, took refuge (4th cent) on Mt. Titano, the chief geographical feature of present-day San Marino. By the mid-5th cent., a community was formed; because of its relatively inaccessible location and its poverty, it has succeeded, with a few brief interruptions, in maintaining its independence. In 1631 its independence was recognized by the papacy."*
Other Answer(s)	*"In addition to the "real" nations, there are so-called "micronations" which arise when someone claims that their little chunk of land is an independent country. A good example is Sealand, a man-made structure a few miles off the coast of England once used as a lookout post during WWII. Someone took over the platform, declared it to be the Principality of Sealand, and declared himself king. Sealand is about the size of an oil drilling rig."*
Category	Geography
MainCategory	Science Mathematics

linek and Mercer, 1989) was used to interpolate our extended models with the baseline model.
To use the interpolation of the baseline and the class-based model, the probability q_i given both word- and class-based models is computed from (3) and (4) and interpolated by a weighting parameter λ.

$$P(Q|S) = \prod_{i=1...M} [\lambda\, P(q_i|C_{q_i}S)P(C_{q_i}|S) + (1-\lambda)\, P(q_i|S)] \tag{8}$$

The similar interpolation model can be use for combining the baseline and the trained trigger model in which Equations 3 and 6 are used.

$$P(Q|S) = \prod_{i=1...M} [\lambda \frac{1}{N} \sum_{j=1...N} P(q_i|s_j) + (1-\lambda)\, P(q_i|S)] \tag{9}$$

The linear interpolation of all models is calculated as follows:

$$P(Q|S) = \prod_{i=1...M} [\lambda_1\, P(q_i|C_{q_i}S)P(C_{q_i}|S) + \lambda_2 \frac{1}{N} \sum_{j=1...N} P(q_i|s_j) + (1-\lambda_1-\lambda_2)\, P(q_i|S)] \tag{10}$$

For the class-based model, we need to define number of clusters that the vocabulary should be partitioned into. Inasmuch as the optimum number of clusters to be used in the Brown algorithm is not self-evident, tests were conducted with several numbers of clusters. Figure 2 shows the mean average precision (MAP) of the class-based unigram for varying numbers of clusters. Setting the number of clusters to 0, the model works like the uniform model. Equating the number of clusters with the number of words is equivalent to using a word-based model. According to the results, the best mean average precision is achieved by clustering all 34,000 lexical items into 200 clusters. Hence, this value was used in all further experiments.

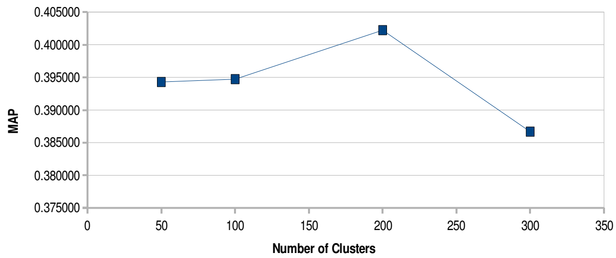

Figure 2: MAP of the class-based model over different numbers of classes

5.4. Results

Table 3 shows the results of our experiments for the the pure class-based and pure trained trigger models in which Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision at 5 (P@5) serve as the primary metrics. Comparing the results to the baseline, we can see that the trained triggering model performs very poorly, while the MAP and P@5 in the class-based model are similar to the baseline and the class-based model outperforms the baseline in MRR.
As we expected, each of the proposed models can not perform accurately when applying individually. Hence, in the second step we used the interpolation of our models with the baseline. The results are presented in Table 4 while the interpolated models are compared with the baseline model. From this table, we observe that although the trained triggering model performed very poorly, interpolating this model with the baseline improves the sentence re-

Table 3: Retrieval results with different language modeling schemas

Language Model	MAP	MRR	P@5
Baseline Model	0.3701	0.5047	0.2267
Class-based Model	0.3705	0.5239	0.2233
Triggering Model	0.0344	0.0415	0.0099

Table 4: Retrieval results for the linear interpolation of proposed models with the baseline

Language Model	MAP	MRR	P@5
Baseline Model	0.3701	0.5047	0.2267
+ Class-based Model	0.3876	0.5368	0.2390
+ Triggering Model	0.4371	0.5655	0.2628
+ Class-based & Triggering	0.4415	0.5729	0.2645

trieval performance significantly. Interpolating class-based model with the baseline also improved the system performance, but the improvement was not as pronounced with the triggering model. We also interpolated all the three models, the baseline, the class-based, and the trained triggering models together and achieved another step improvement on the system performance.

Figure 3 shows the precision-recall curve of the baseline model and all of the interpolation models. This curve indicates the superiority of our proposed model to the baseline model such that the proposed model significantly outperform the baseline at the level of $p-value < 0.01$ according to the two-tailed paired *t*-test.

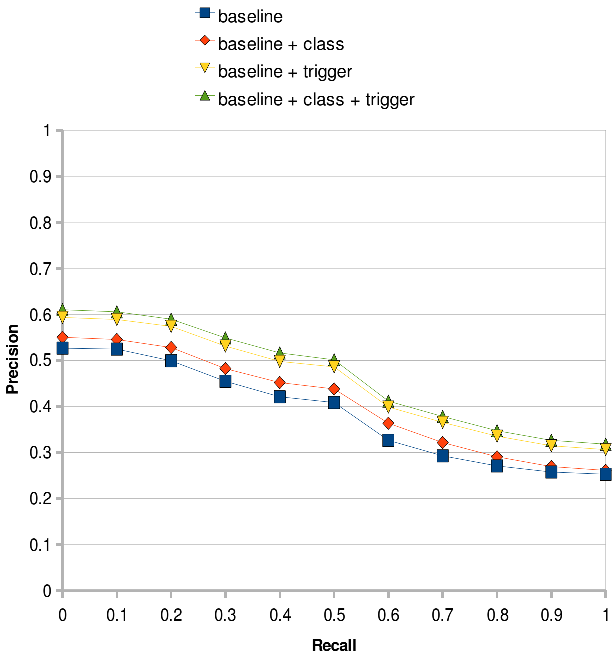

Figure 3: Comparing the precision-recall curve of the baseline model with the interpolated proposed models

6. Conclusion

In this paper, we proposed a way of exploiting the logs derived from community-generated question answering forums in automatic question answering systems to offer more accurate answers to users' questions. To this end, we used the Yahoo! Answer Corpus derived from *http://answers.yahoo.com/* as a community question answering log to retrieve more relevant sentences in an automatic question answering system. The retrieved sentences can be further processed using a variety of information extraction techniques to find the final answers.

Two different language model-based frameworks have been introduced here and trained on the Yahoo! Answer Corpus. Our experiments on TREC question answering track verified that both of the models can improve the sentence retrieval performance, in which interpolating both proposed models with the baseline performs the best compared to each of the individual models.

One possible approach to expand the current model is benefiting from the meta data that Yahoo! provides for this corpus. At the moment, no meta information is used in our model. However, it is probable giving a higher priority to the best answer labeled in the corpus or using the category of the question improves the model.

Acknowledgments

The authors would like to thank Grzegorz Chrupala for interesting discussions. We, in particular, thank Yahoo! Labs Webscope for their Yahoo! Answers Comprehensive Questions and Answers corpus. Saeedeh Momtazi is funded by the German research foundation DFG through the International Research Training Group (IRTG 715) Language Technology and Cognitive Systems.

7. References

A. Berger and J. Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of ACM SIGIR International Conference*, pages 222–229.

P.F. Brown, J. Cocke, S.D. Pietra, V.J Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

P.F. Brown, V.J.D. Pietra, P.V. Souza, J.C. Lai, and R.L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

C. Burgess, K. Livesay, and K. Lund. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes*, 25(2,3):211–257.

G. Cao, J.Y. Nie, and J. Bai. 2005. Integrating word relationships into language models. In *Proceedings of ACM SIGIR International Conference*.

G. Cao, J.Y. Nie, and J. Bai. 2006. Constructing better document and query models with markov chains. In *Proceedings of ACM CIKM International Conference*.

E. Charniak. 1993. *Statistical Language Learning*. The MIT Press, Cambridge MA.

C.Y. Chung and B. Chen. 2002. CVS: a correlation-verification based smoothing technique on information retrieval and term clustering. In *Proceedings of ACM SIGKDD International Conference*.

W.B. Croft and X. Wei. 2005. Context-based topic models for query modification. Technical report, University of Massachusetts, Amherst.

H.T. Dang, J.J. Lin, and D. Kelly. 2006. Overview of the trec 2006 question answering track. In *Proceedings of Text REtreival Conference (TREC)*.

J. Gao, J.Y. Nie, G. Wu, and G. Cao. 2004. Dependence language model for information retrieval. In *Proceedings of ACM SIGIR International Conference*.

D.A. Hiemstra. 1998. Linguistically motivated probabilistic model of information retrieval. In *Proceedings of European Conference on Digital Libraries*, pages 569–584.

F. Jelinek and R. Mercer. 1989. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of an International Workshop on Pattern Recognition in Practice*.

Fred Jelinek. 1998. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge MA.

K.S. Jones. 1971. *Automatic Keyword Classification for Information Retrieval*. Butterworths, London.

M. Kaisser and J.B. Lowe. 2008. Creating a research collection of question answer sentence pairs with Amazon's mechanical turk. In *Proceedings of the LREC International Conference*.

J. Lafferty and C. Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of ACM SIGIR International Conference*, pages 111–119.

X. Liu and W.B. Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of ACM SIGIR International Conference*, pages 186–193.

S. Liu, F. Liu, C. Yu, and W. Meng. 2004. An effective approach to document retrieval via utilizing WordNet and recognizing phrases. In *Proceedings of ACM SIGIR International Conference*, pages 266–272.

D.J.C. Mackay and B. Peto. 1995. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):1–19.

R. Mandala, T. Tokunaga, and H. Tanaka. 1998. Ad hoc retrieval experiments using wordnet and automatically constructed theasuri. In *Proceedings of Text REtreival Conference (TREC)*, pages 475–481.

C.D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

A. Merkel and D. Klakow. 2007. Comparing improved language models for sentence retrieval in question answering. In *Proceedings of Computational Linguistics in the Netherlands Conference*, pages 475–481.

D. Miller, T. Leek, and R.M. Schwartz. 1999. A hidden markov model information retrieval system. In *Proceedings of ACM SIGIR International Conference*, pages 214–222.

S. Momtazi and D. Klakow. 2009. A word clustering approach for language model-based sentence retrieval in question answering systems. In *Proceedings of ACM CIKM International Conference*, pages 1911–1914.

S. Momtazi, S. P. Khudanpur, and D. Klakow. 2010. A comparative study of word co-occurrence for term clustering in language model-based sentence retrieval. In *Proceedings of NAACL International Conference*.

V.G. Murdock. 2006. *Aspects of sentence retrieval*. Ph.D. thesis, University of Massachusetts Amherst.

R. Nallapati and J. Allan. 2002. Capturing term dependencies using a sentence tree based language model. In *Proceedings of ACM CIKM International Conference*, pages 383–390.

J. Ponte and W.B. Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of ACM SIGIR International Conference*, pages 275–281.

Y. Qui and H. Frei. 1993. Concept based query expansion. In *Proceedings of ACM SIGIR International Conference*, pages 160–169.

C.J. Van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London, 2 edition.

S. E. Robertson, C.J. van Rijsbergen, and M.F. Porter. 1981. Probabilistic models of indexing and searching. In *Proceedings of ACM SIGIR International Conference*, pages 35–56.

H. Schütze and J.O. Pedersen. 1997. A cooccurrence-based thesaurus and two applications to information retrieval. *Information Processing and Management*, 33(3):307–318.

D. Shen. 2008. *Exploring Rich Evidence for Maximum Entropy-based Question Answering*. Ph.D. thesis, Saarland University.

F. Song and W.B. Croft. 1999. A general language model for information retrieval. In *Proceedings of ACM SIGIR International Conference*, pages 279–280.

A. Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the Spoken Language Processing Conference*.

M. Surdeanu, M. Ciaramita, and H. Zaragoza. 2008. Learning to rank answers on large online QA collections. In *Proceedings of ACL International Conference*.

T. Tao, X. Wang, Q. Mei, and C. Zhai. 2006. Language model information retrieval with document expansion. In *Proceedings of NAACL-HLT International Conference*, pages 407–414.

E. Voorhees. 1994. Query expansion using lexical-semantic relations. In *Proceedings of ACM SIGIR International Conference*, pages 61–69.

Yahoo! Labs Webscope. 2009. Yahoo! answers comprehensive questions and answers [version 1.0]. Website. `http://webscope.sandbox.yahoo.com`.

X. Wei and W.B. Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of ACM SIGIR International Conference*, pages 178–185.

X. Wei and W.B. Croft. 2007. Modeling term associations for ad-hoc retrieval performance within language modeling framework. In *Proceedings of European Conference of Information Retrieval*, pages 52–63.

(Tacitly) Collaborative Question Answering Utilizing Web Trails

Sharon G. Small[1] **and Tomek Strzalkowski**[1,2]
[1]The Institute for Informatics, Logics and Security Studies
The State University of New York at Albany
Albany, New York 12222, USA
[2]Institute of Computer Science, Polish Academy of Sciences
E-mail: small@albany.edu, tomek@albany.edu

Abstract

In this paper we explore the concept of tacitly collaborative question answering – where the system enables users to indirectly share and exchange relevant knowledge without requiring dedicated effort on their part. Our aim is to exploit the exploratory knowledge that Internet users create each time they search for information on the web. If this exploratory knowledge can be captured and distilled in a systematic way, and then efficiently shared with the subsequent users, the effect would be that of collaborative question answering – a vast improvement over the current web search paradigm. In this paper we describe preliminary experiments focused on capturing users' web exploration trails and determining the degree to which these trails overlap among the users who are seeking similar information. We recorded 95 distinct exploration trails from 11 search experiments, with 4-8 users per task, where the users were given the same information tasks to solve but not how to approach them. We found that for more than half of these users their exploratory trails overlap to a significant degree (> 50%) indicating that these users could directly benefit from collaborative knowledge sharing by being offered informed search shortcuts – a form of community based relevance feedback.

1. Introduction

Users who are looking for answers and solutions on the web leave behind trails of their exploratory activities that contain specific, practical knowledge – how they did it, what queries did they ask, what links they tried, what worked and what did not. This discrete, episodic knowledge (e.g., Tulving, 1972; Braumann et al., 1995; Najjar & Mayers, 2003) forms a layer of metadata that supplies meaning to the underlying information; it may clarify or even alter the meaning that was originally intended for any posted data item. For example, a resourceful user may discover that a certain product offered for sale (e.g., wine cooler) can be adapted to storing apples by replacing the wire shelving inside. Another user, facing a similar problem may be able to find the answer faster and with less effort, if only he or she could utilize this knowledge that the first searcher just discovered. If such exploratory knowledge can be captured and distilled in a systematic way, the effect would be that of collaborative and interactive question answering/problem solving, where web searchers tacitly collaborate in building a better resource for themselves. We recorded 95 distinct exploration trails from 11 search experiments, with 4-8 users per task. We found that for more than half of these users their exploratory trails overlap to a significant degree (> 50%) indicating that these users could directly benefit from collaborative knowledge sharing by being offered informed search shortcuts.

2. Related Research

The preliminary work described in this paper bears some relation to the technologies underlying social networking as well as work aimed at constructing web-based ontologies. Exploiting the full potential of the web as a communication, information, and commerce medium requires vastly improved human-information interaction methods. Current Internet search technology is inadequate because of limited means to understand the user needs as well as the meaning of information being accessed. Various ontology-based proposals to standardize web content (e.g., the "semantic web") are both expensive and impractical because they depend crucially on universal adherence to agreed upon standards (e.g., Shadbolt et al., 2006). Various forms of human-built ontologies, such as Wordnet (Fellbaum, 1998), FrameNet (Baker et al., 2003) Verbnet (Kipper et al., 2002), Cyc (Lenat, 1995) etc. have limited usefulness due to their rigid structure and designer biases; except for limited cases (Moldovan et al., 2008) they were found to be ineffective in real applications (e.g., IR Voorhees 2003; QA Prager et al., 2006). Some applications utilize social tagging of web content (such as del.icio.us) in order to create informal, user-driven ontologies (or *folksonomies*), but their objectives are usually to share and exchange items based on their popularity rather than utility for any particular purpose. Another form of socially-enhanced search, known as collaborative filtering, is an attempt to profile the user, which may raise privacy issues while also being generally hard to achieve, except in narrowly defined domains (e.g., Amazon.com book recommender).

In this paper we explore the possibility that such shared knowledge may be possible to obtain without dedicated effort of Internet users, or teams of experts. It turns out that a majority of informal encodings that people associate with data items (texts, images, videos, maps, games, products, etc.) are created (if only virtually) for a particular purpose rather than for "general" community reference. They are created for the convenience of the annotator and may reflect a specific task or objective that the person is pursuing (e.g., Maier and Delcambre, 1999).

A good example of such virtual annotation is a web search query: it is an explicit request to locate data items that posses a specified set of characteristics. The returned results that are judged by the searcher as particularly relevant (e.g., by copying, or saving, or purchasing something) may be assumed to meet these criteria. This is roughly equivalent to folksonomic annotation, except that the objects are "encoded" for their meaning and role with respect to the user's task, and not for their more objective qualities or for sharing. In many situations, additional browsing and follow-up queries would provide an even richer context. At the same time, since the encodings are "private", and thus not meant to be seen by anyone else, they may be considered more "honest" than explicit folksonomic annotations where a community "image" is a factor. Most importantly: such virtual annotations are created continuously in vast quantities by many users who repeatedly retrieve and touch the same data items, and these annotations tend to be quite accurate since the users are highly motivated.

3. Collaborative Knowledge Layer from Multiple Web Trails

The research we are advancing is radically different from how information search is done today: it ends the clear-cut separation between the information producers and offerers, on the one hand, and the users and consumers, on the other. The users who are looking for answers and solutions leave behind specific, practical knowledge – how they did it, what queries did they ask, what worked and what did not, i.e. a web trail. This discrete, episodic knowledge forms a Collaborative Knowledge Layer of metadata that supplies meaning to the underlying data; it may clarify or even alter the meaning that was originally intended. Utilizing this knowledge from web trails may result in improved interactive question answering that is also indirectly (tacitly) collaborative: each time one searcher solves a complex search problem, the solution, its elements, and generalizations may now become available to subsequent searchers. This may be schematically illustrated by the drawing in Figure 1 below, where U_n represent users, Q_n represent questions they pose, and F_n represent certain other exploratory moves (decisions, actions) they make about data items found.

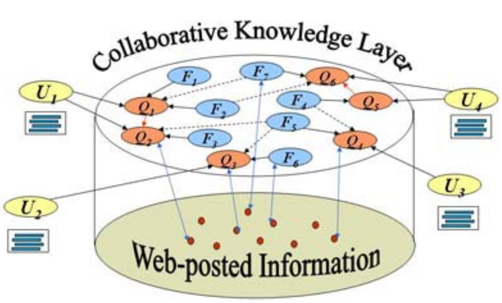

Figure 1: Intersecting Web Trails

The objective is to (a) capture web exploration trials and (b) distill them into knowledge elements that can be used by a search system to steer subsequent searchers faster towards their intended goals. One way to achieve the latter is by adding a meaning layer to the underlying web data (e.g., that 'wine cooler' can be a partial answer to 'apple storage' question.). In this paper we focus primarily on (a). We should note that our approach is distinct from existing social web tools; it does not require users to perform any explicit decision or labeling actions. All user actions are automatically captured, and this is not limited to successful outcomes only.

4. Exploratory Episodes

A web trail consists of a sequence of individual exploratory moves that may include activities such as: entering a search query, typing other text into an input box, responses from the browser, and any offers accepted or ignored, e.g. files saved, items viewed, links clicked through, etc. Within such trails, there will be subsequences that are optimal paths leading to specific outcomes – common portions of exploratory trails that multiple network users already traversed. We will isolate these subsequences as exploratory episodes that may be shared with any new users who appear to be pursuing a compatible task, i.e. asking a similar question or viewing the same items, etc. All exploratory episodes would be stored and would eventually form clusters based on degrees of mutual overlap (actual and semantic). Divergent episodes would not be discarded but would be stored as new knowledge to be shared when a sufficiently close exploration path by another user is detected.

Figure 2 below shows a simplified graphical rendering of a discovered exploratory episode and how it can be leveraged to assist users during an interactive question answering session. A more expanded, detailed version of a single user exploratory episode typical of our experiments can be seen in Section 6.

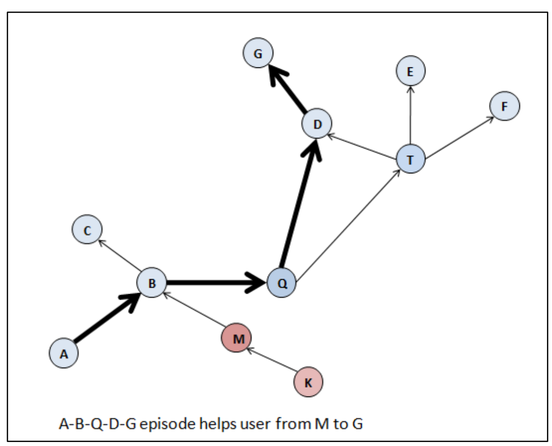

Figure 2: Discovered overlapping trails assist new user.

In Figure 2, nodes indicate locations in the network, which may be data items (A, B, etc.) or queries that are entered (Q). Links indicate the direction of navigation, via explicit links, URL addresses, or by obtaining results from a search. For example, the query submitted at Q returns a page T of links, which may now be explored one by one (D, E, F), but only one is ultimately followed (D). This fictional exploration ends at node G, but it is clear that an optimal route would be A-B-Q-D-G, and this trail would become an exploratory episode. The episode thus discovered may have been created by a single user; however, its importance is increased if more users have followed the same trail, whether on their own or by being helped by the system. For example, a new user starting a trail K-M reaches B at which point he or she might be "offered" to go directly to G, especially if the next move were to enter query Q. The shortcut offer represents the system's understanding of what the user may be looking for. If the new user accepts the offer, it may represent a substantial savings not only in the length of the navigation path, but more significantly, in time and effort that the user needs to invest in the search. In other words, the knowledge vested in the system by users who already went through the same process is now returned to help the new user along.

5. Experimental Data Collection

In order to demonstrate feasibility of our exploratory episode's concepts, we collected data from a series of workshops (Strzalkowski et al., 2007, 2006) conducted with several groups of professional analysts using the HITIQA [1] system (Small, 2007) and a collaborative extension known as COLLANE (Strzalkowski, et al., 2009). During these workshops the analysts were presented with a selection of research problems and asked to prepare draft reports on each problem. Here are a few examples of the topics assigned:

Artificial Reefs

Many countries are creating artificial reefs near their shores to foster sea life. In Florida a reef made of old tires caused a serious environmental problem. Please write a report on artificial reefs and their effects. Give some reasons as to why artificial reefs are created. Identify those built in the United States and around the world. Describe the types of artificial reefs created, the materials used and the sizes of the structures. Identify the types of man-made reefs that have been successful (success defined as an increase in sea life without negative environmental consequences). Identify those types that have been disasters. Explain the impact an artificial reef has on the environment and ecology. Discuss the EPA's (Environmental Protection Agency) policy on artificial reefs. Include in your report any additional related information about this topic.

Effect of Focused Vibrations on the Human Brain

Please write a report on the effects of whole body vibration (WBV) of limited duration on the human brain. Include in your report the current state of science (and general consensus) with respect to man's ability to tolerate vibrations at various levels of exposure. Discuss claims that focused vibrations can heal or strengthen muscle tissue in parts of the body. Does using a jackhammer strengthen a person's arm muscles? Also report on whether there is a definite limit at which man's exposure to WBV is known to cause injury. Include any other important information.

Islamic Movement of Uzbekistan (IMU)

After government forces raided and killed 19 Islamic militants in Tashkent, Uzbekistan, the Secretary of Defense has requested a briefing on the terrorist group - Islamic Movement of Uzbekistan (IMU). What other possible names is the IMU known by? What is its ideology and political goals? Where was the group started and who are its founders? Who are the leaders of this organization? What other modern terror groups are connected to the IMU? What is the current estimated strength of the IMU? Where in Central Asia can cells of this group be found? What terrorist operations have the groups executed in the last five years? What were the impacts of these actions?

Each analyst used the system for a preset time limit, approximately 2.5 hours, to collect and organize information, to prepare a draft report. Each analyst had access to the same dataset: a 2 GB corpus of web-mined text.

The data collection to support the experiment was assembled ahead of time by mining the open web for information related to the topics selected for the workshop. The data was mined by breaking topic descriptions into approximately one hundred short search queries. For each query, the top 500 Google results were retained. This added up to 50,000 URLs, initially producing about 500 MB for each topic. The retrieved web pages were filtered and stripped of XML to retain only the well-formed text sources, and also to eliminate meaningless, commercial or offensive content. The resulting data set was sufficiently varied to allow meaningful exploration. An important consideration here was to obtain a stable dataset for the duration of the experiment; since our objective was to collect user actions we needed to control the number of variables in order to obtain meaningful results.

All activities of the analysts were recorded. The following is a partial list of key logged events.

Examples of tracked system events:

- Questions Asked
- Data Items copied
- Data Items ignored
- System offers accepted and/or rejected
- Displaying text
- Word(s) searched on the user interface
- All dialogue between the user and the system
- Bringing up a full document source
- Document collection selected
- Passages viewed
- Time spent

[1] HITIQA is the Question Answering Research Prototype system developed under the AQUAINT program at the State University of New York at Albany.

Although the purpose of these workshops was to evaluate effectiveness of the question answering technology, in terms of answer accuracy, report quality, and user satisfaction, the experiments yielded an extremely rich corpus of exploratory data. The items listed above were automatically captured into each analyst's personal log, along with time stamp information. In particular, the logs captured detailed web trails that each analyst took while searching through the data.

While each analyst had access to the same data repository (simulating a corporate intranet) and they were given the same problems to research, they could not communicate otherwise. The objective of the search was to find sufficient information for a 3-page report by copying passages from documents found on the network. The report consisted of verbatim citations (excerpts) from source documents (along with references) and analysts' own commentary; however, all conclusions had to be explicitly supported by cited evidence. We can therefore consider the data items from which some evidence was copied as relevant to the task, and thus part of an exploratory episode. We wanted to observe, how many users would follow similar search paths, and for those who did, how much of their exploratory episodes overlapped. Specifically, we wanted to know if they touched the same data items (documents) and when they did, whether they used them in the same way. Moreover, if they indeed fell in step, did they continue along the same path? In this paper our focus is on the degree of overlap of copied data items between any two analysts.

6. Experimental Results

The experimental conditions under which this data has been collected differ somewhat from the open web search, and are more typical of navigation of corporate internal data networks. In particular, users navigated the data through a custom GUI which allowed them to enter queries and to display search results, but not to navigate away from these. Thus, a navigation path repeatedly returns to the query node, creating a distinctive clustered path as shown in Figure 3.

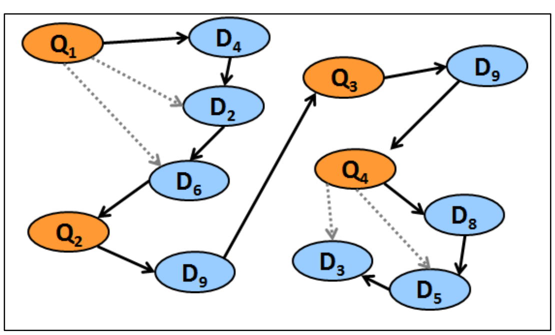

Figure 3: A Single User Exploratory Episode consists of nodes linked by solid arrows. Dotted arrows indicate data items returned in response to a query but not immediately examined by the user.

The exploration path in Figure 3 starts with a user entering query Q_1 and then reviewing data items returned (D_4, D_2, D_6). Next, the user proceeds to query Q_2 and so on. In this graph, only the data items that were considered relevant (i.e., from which evidence was copied by the user) are shown – in other words, Q_1-D_4-D_2-D_6-Q_2-D_9-… is an exploratory episode obtained.

In these experiments the users were working on the same task; nonetheless their exploratory paths often looked quite different at first glance. Different users took a variety of detours and asked their questions in a different order – still many of them ended up collecting very similar sets of evidence. In order to see how much their exploratory episodes overlapped, we needed to un-scramble them and remove what could be considered fruitless detours. To do so, we created "inverted episodes" by imposing a single order upon all data items and then aligning the episodes by the common data items they contained, rather than being ordered sequentially by the time the document was visited by each individual analyst. As explained above, in our initial analysis we only considered user copy actions as indicators that particular data items were useful. We reasoned, that if two users arrived at the same answer following differently ordered paths they could also be steered to follow a somewhat different route and still arrive at the same result, though perhaps through a more efficient route. The inverted episodes are easily represented in a tabular form as shown in Table 1, where each column represents an episode of a single user.

	User A	User B	User E	User F	User G	User H
doc_006449		Q6	Q5	Q7		Q4
doc_002135	Q1		Q5		Q2	Q7
doc_005860			Q4	Q3		Q3
doc_001221		Q1	Q8			
doc_003507				Q3	Q6	Q3
doc_001124				Q4	Q8	
doc_002598				Q4		Q8
doc_004684	Q1		Q4		Q2	Q3
doc_005866			Q8		Q8	
doc_004730				Q4	Q8	
doc_005286	Q1		Q4	Q1	Q2	
doc_005907					Q6	Q3
doc_002339	Q1		Q5		Q7	Q7, Q8

Table 1: Data items found and copied by two or more users for the Artificial Reefs topic.

Table 1 above shows the aligned partial inverted episodes generated by six users researching the same topic: Artificial Reefs. The leftmost column in the chart shows all data items (documents) that were found and copied from by at least 2 users.

We note, for example, that users A and G as well E and H display a remarkable degree of overlap in their explorations – by selecting the same data items 60-75% of the time. This overlap is very high when we consider that the general inter-judge agreement observed in document relevance assessments is at about 70-80% (Voorhees, 2005); moreover, we must take into account that this overlap was measured strictly by information item id, and not by their content. This chart therefore represents a lower bound on episode overlap.

The chart also shows that users A and B appear to follow radically different paths through the data. Similarly, there are more differences than similarities between the exploratory episodes generated by users E and F. Assuming that these divergences are genuine (i.e., ignoring potential content overlap between the different copied items), we may stipulate that these users would not benefit from information sharing between them as much as A and G or E and H would. This may be because A and B were looking for very different solutions to the same problem. In general, we noted that exploratory episode overlaps tend to be either high (i.e., 60% or higher) or low (i.e., 40% or less), which suggests the existence of an overlap threshold above which information sharing should be attempted.

While Table 1 represents a fairly typical case seen in our experiments, we also found more extreme situations on both ends of the spectrum. Table 2 shows the inverted episodes obtained from a group of 4 users who researched the topic of Islamic Movement of Uzbekistan. In their relatively shorter explorations two of the participants (users A and C) showed a near 100% episode overlap, while their episodes had almost nothing in common with User D – in fact User D found almost only unique data items that no other user had considered.

	User A	User B	User C	User D
doc_071947	Q1		Q1	
doc_211286	Q1		Q1	
doc_090493				Q6
doc_060698				Q1
doc_051659	Q1	Q1	Q1	
doc_071951			Q1	
doc_059950				Q5
doc_208982	Q1	Q1	Q1	Q1
doc_160802	Q1	Q2	Q1	

Table 2: Data items found and copied by users for the IMU topic.

Table 3 below shows the overlap analysis for 95 exploratory episodes obtained from 11 search experiments, with 4-8 users per task. The two sides of the table assume a different cross-episode overlap threshold before attempting episode sharing between users. The numbers under the "Low" and "High" columns represent the number of pairs of analysts that were above the overlap threshold: "High", or below the overlap threshold: "Low". For example, the Artificial Reefs topic had 11 pairs of analysts with greater than or equal to 50% overlap in their data items copied, and 4 pairs that had less than 50% overlap in their data items copied.

	Threshold => 50%		Threshold => 60%	
	Low	High	Low	High
Biological Warfare	0	6	1	5
Burmese	4	2	2	4
Islamic of Uzbekistan	0	6	0	6
Terror Attack	5	1	5	1
Spain	1	5	4	2
Vibrations	8	2	9	1
Artifical Reef	4	11	8	7
HoneyBee	3	3	4	2
Teflon	3	3	5	1
Tainted Food	6	9	10	5
Software Piracy	8	5	11	2
Total	**42**	**53**	**59**	**36**
% of High Overlap		**55.79%**		**37.89%**

Table 3: Summary of episode overlap statistics computed over 95 exploratory episodes obtained from 11 intranet search experiments

At a 50% episode overlap threshold, more than half of all episodes are candidates for sharing, on average; at 60% episode overlap, slightly more than a third of searches are sharable. We also note that some of the topics are more suitable for information sharing and tacit collaboration among the users, specifically these topics where the exploratory episodes tend to overlap highly for most users (Biological War, Spain, IMU and Artificial Reefs). Similar statistics can be obtained for higher or lower thresholds (70%, 40%, etc.). We assume the higher the overlap requirement the more likely the shared episode will be accepted by the user. This will be explored in our future research.

7. Collaborative Sharing

Our objective is to leverage this exploratory knowledge to provide superior accuracy and responsiveness of the question answering system. What we leverage here is the experience and judgment of other users who faced the same or similar problem before, and we now turn their experience – and the resulting exploratory knowledge – into an additional source of evidence. The effect is similar to that of relevance feedback in information retrieval, except that the exploratory knowledge is community based rather than depending upon introspection of a single user. In this way we hope to transform a merely vast networked information repository (which is the Internet, by and large) into a self-sustaining knowledge resource.

To illustrate collaborative sharing and the potential performance improvement we provide an example from one of our experiments in Figure 4 below.

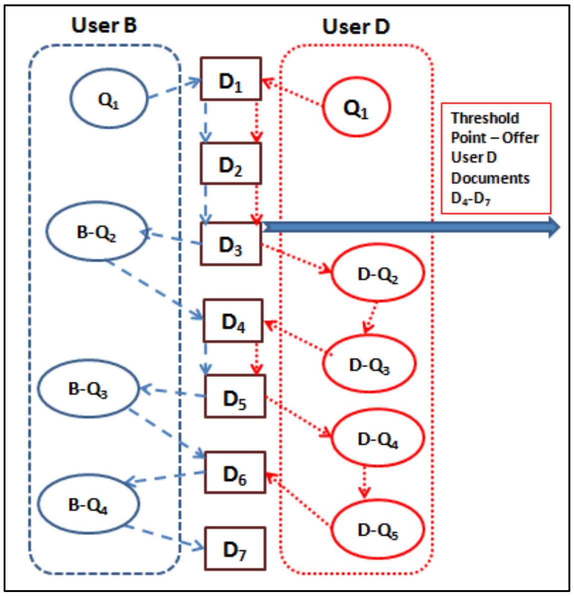

Figure 4: Actual interaction with the system for User B and User D. User B Exploratory Episode may be shared to assist User D.

Illustrated in Figure 4, are the exploratory trials by two users B and D. Both users attempt to solve the same information problem, but they approach it differently. When the two are working independently of one another, they traverse the paths as shown in both sides of Figures 4. User B starts with query Q_1, followed by copies from documents D_1-D_3 and then another query B-Q_2, and more copies from D_4, etc. In the proposed system this history is captured and saved as an exploratory episode. Subsequently, User D enters the system and begins along the same trail, asking exactly the same question, Q_1, and copying from the same documents D_1-D_3. At this point we have a 100% overlap between User D's work and the exploratory episode of User B. This of course is well above even our conservative 60% threshold, and we can feel confident in offering User D documents D_4-D_7 as shown in the exchange in Figure 5 below. This would in essence give him all the data he would have eventually found but not without 4 more questions to the system, 2 of which are fruitless with nothing copied. Additionally, User D would be given a data point, D_7, that he may not have found on his own and that has a high probability of being relevant given his significant overlap to User B's actions.

Other users entering the system and following this exploratory episode with a 60% or higher overlap would also be offered the appropriate documents from User B's path. That is, documents they had not yet seen nor explicitly asked for yet, but that had been deemed relevant by User B's copies. Users below the threshold would not trigger this episode and therefore would not be offered the documents from User B's exploratory path.

User D: Q_1: *What type of toxins are Anthrax, Botulism and Smallpox?*

[System displays answer passages from a set of documents including D_1-D_3.]

[User D copies from D_1-D_3]

User D: D-Q_2: *What type of toxins are in Anthrax?*

[System displays answer passages for question D-Q_2.]

System: *You may also find the following information useful:* [Displays answer passages D_4-D_7].

Figure 5: Possible interaction with User D utilizing the Exploratory Episode captured from User B. (Actual questions from User D extracted from experiment logs.)

8. Conclusion

We have demonstrated the feasibility of utilizing search trails to improve effectiveness of question answering:

1. Users searching for information in a networked environment leave behind an exploratory trail that, along with their decisions about all information items viewed, form an exploratory episode that characterizes the search just completed. These episodes can be captured, stored and compared against one another.
2. Exploratory episodes can be robustly compared for overlap by keeping track of data items users view and make decisions about as they search for information on the web.
3. Many users searching for the same or highly related information are likely to follow similar routes through the data, but their paths may not all be optimal. When their paths show overlap above a certain threshold, they are likely searching for the same information – these users will benefit from tacit information sharing technology.

9. Future Research

Our future work will first focus on evaluating the overlap of data items copied when we consider the semantic equivalence of their content rather than simply information item id. We will also be interested in distilling exploratory episodes into knowledge elements that can be offered to users found to be following a similar trail. Along this line we will develop metrics to automatically judge whether two users are indeed following a similar trail. These metrics will include not only data items copied, but we will expand this comparison to include other key activities: e.g. similarities of questions asked, items ignored, etc. Finding the right threshold that balances potential benefits to the user and the likelihood of acceptance of a system's offer will be one of the key tasks in our continued research. Finally, we will evaluate how often offered material is accepted by users and to what degree this speeds up their performance and improves their satisfaction with the search tool.

10. Acknowledgements

This research was funded in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) under the CASE Program, through an award to the University at Albany. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI or the U.S. Government.

This material is based in part upon work supported by the National Science Foundation under Grant No. 0810404. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

11. References

Baker, C., Fillmore, C., & Cronin, B. (2003). The Structure of the FrameNet Database, International Journal of Lexicography, Volume 16.3:281-296.

Braumann, U-D., Boehme, H-J., and Gross, H-M. (1995). An Episodic Knowledge Base for Object Understanding. *Proceedings of 3rd ESANN*, Brussels.

Fellbaum, C., editor. (1998). Wordnet An Electronic Lexical Database. MIT Press, May 1998.

Kipper, K., Palmer, M., & Rambow, O. (2002) Extending PropBank with VerbNet Semantic Predicates. Workshop on Applied Interlinguas, held in conjunction with AMTA-2002. Tiburon, CA.

Lenat, D. (1995). “Cyc: A Large-Scale Investment in Knowledge Infrastructure.” Communications of the ACM 38, no. 11.

Maier, D. & Delcambre, L. (1999) “Models for Superimposed Information.” Proceedings of Workshop on Conceptual Models for the WWW in conjunction with the International Conference on Conceptual Modeling, (ER ’99), November, 1999, Paris, France, Lecture Notes in Computer Science 1727, Springer Verlag, pp 264-280

Moldovan, D., Pasca, M., and Surdeanu, M. (2006). Some Advanced Features of LCC’s Power Answer, in Advances in Open-Domain Question Answering, (T. Strzalkowski & S.Harabagiu eds.) pp. 3-34. Springer.

Najjar, M., Mayers, A. (2003). A Computational Cognitive-Based Approach to Represent Knowledge within Intelligent Tutoring Systems. *Proceedings of The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT) 2003*, Athens, Greece.

Prager, J. M., Chu-Carroll, J., Brown, E. & Krzysztof C. (2006). Question Answering by Predictive Annotation, in Advances in Open-Domain Question Answering, , (T. Strzalkowski & S.Harabagiu eds.) pp. 307-347. Springer.

Shadbolt, N, W. Hall, and T. Berners-Lee (2006). Semantic Web Revisited. IEEE Transactions on Intelligent Systems, May-June 2006. pp. 96—102. Semantic Web refs

Small, S. (2007). An Effective Implementation of Analytical Question Answering. The State University of New York at Albany, Ph.D. Thesis.

Strzalkowski, T., Taylor, S., Shaikh, S., Lipetz, B-A., Hardy, H. Webb, N., Cresswell, T., Wu, M., Zhan, Y., Liu, T., & Chen, S. (2009). COLLANE: An experiment in computer-mediated tacit collaboration. In *Aspects of Natural Language Processing* (M. Marciniak and A. Mykowiecka, editors). Springer.

Strzalkowski, T., Shaikh, S., Small, S., Hardy, H., Webb, N., Wu, M., Liu, T., and Zhan, Y. (2007). COLLANE: The Collaborative Analytical Environment. DTO/IARPA Technical paper.

Strzalkowski, T., Small, S., Taylor, S., Lipetz, B., Hardy, H., and Webb, N. (2006). Analytical Workshop with HITIQA. Report submitted to the Disruptive Technologies Office.

Tulving, E. (1972). Episodic and semantic memory. In *Organization of memory* (Tulving & Donaldson eds.). Academic Press, New York.

Voorhees, EM; Harman, DK. (2005). TREC: Experiment and Evaluation in Information Retrieval. Cambridge, MA, MIT Press.

Voorhees E. (2003). Overview of TREC 2003. Proceedings of the Text Retrieval Conference, TREC 2003.

Named Entity Recognition in an Intranet Query Log

Richard Sutcliffe, Kieran White, Udo Kruschwitz

Department of Computer Science
University of Limerick, Ireland

School of Computer Science and Electronic Engineering
University of Essex, UK

Richard.Sutcliffe at ul dot ie, udo at essex dot uk dot ac

Abstract

A Named Emtity is a piece of information of a particular type such as a person name, company name, a place, a time or date, and so on. We have observed that NEs occur in intranet search engine logs such as the one at the University of Essex. We first conducted a hand study of a sample of 1794 queries extracted from the log. This allowed us to categorise the queries by topic and hence to determine the most important topics which a search engine in this domain should deal with effectively. In the process, we identified 35 NEs which are important within a university setting. We next carried out a study in which the maximum entropy tagger of the NLPTools package was trained to recognise instances of those NEs in queries. This was a pilot study, but we got good results with very high precision, especially where an NE was clear and where we had sufficient training examples. Recall varied and could be low but this was due to our pilot setup and not due to the algorithm. At the end of the paper we discuss some applications of NE recognition in producing more effiective search engines or question answering engines within a closed intranet domain.

1. Introduction

Following the publication of Jansen, Spink and Saracevic (2000), web log research has become an active area. There are two types of search environment which are of interest. The first is the general internet where users submit queries on any topic to an engine such as Google or Excite. The study just referred to is concerned with this type of environment. However, the second is the intranet where queries are restricted to a particular domain and are asked against a specific and restricted (but possibly extremely large) document collection. Our study is of the latter type and follows on from work by Kruschwitz (2003, 2005) and Kruschwitz, Webb and Sutcliffe (2009).

Our domain is the intranet at the University of Essex at which the search engine is that of Kruschwitz and Al-Bakour (2005). Our study falls at the interesting intersection of log analysis with Question Answering (QA). QA has generated a lot of interest in recent years with tracks firstly at TREC and then at CLEF and NTCIR (TREC, 2010; CLEF, 2010, NTCIR, 2010). In this context, a QA system takes as input a short question in a natural language and produces as output an exact answer, extracted from a document collection (Hirschman and Gaizauskas (2001). A key to QA is the concept of the Named Entity (NE) - an idea which developed out of the MUC evaluations (Grishman and Sundheim, 1996). An NE is a piece of information of a particular type such as a person name, company name, a place, a time or date, and so on. An important finding of the MUC series was that NEs contained very important information which was instrumental in processing or understanding a real-life text such as a newspaper article or an inteligence report. This idea had not come to light previously because people were working on artificial texts or children's stories - material which was considered at that time to be a good testbed for Natural Language Processing (NLP) ideas but was not in fact of any intrinsic interest to any one who might for example wish to purchase and use the resulting NLP system for a commercial purpose. A second important finding from MUC was that it was possible to recognise NEs quite accurately in a text.

Returning to QA, a 'factoid' question of the TREC variety asks something where the answer is an NE. For example the question 'Who is the president of the United States' has an answer of type PERSON (the correct answer returned will depend on the document collection being used). Considering now our intranet search logs at Essex, it is apparent on studying them that users are asking questions as well, even though they may only type in one or two words. Moreover, their questions are usually specific to the domain in question - if they were not, they would be using Google not UKSearch (the name of the Essex search engine). Thirdly, we have observed in our studies that queries very commonly contain NEs. From this we conclude that NEs are an important aspect of query log analysis for intranets.

In the present work, we have firstly carried out an initial manual analysis of a subset of the log, categorising queries by topic type and extracting also other feature data. This analysis has led to the production of a list of specific NEs (SNEs) which occur in the logs. Secondly, we have experimented with the automatic recognition of such SNEs in the logs using a Maximum Entropy Model as manifested in the OpenNLP Tools (OpenNLP, 2010).

The rest of the article is structured as follows. We first outline briefly the characteristics of the log being used. Next we describe the manual analysis we carried out, and the results found. Thirdly, we turn to the initial automatic analysis we have done. After this we discuss the information-providing power of NEs within the context of our log and our domain. Finally, we draw some conclusions from this work.

2. The Log at Essex

The log we are using was collected between 1st October 2006 and 30th September 2007. It shows queries which were submitted to the UKSearch system at University of Essex. This system allows a query to be submitted and in addition to returning a result, it also suggests additional query terms which could be added to the search. A user can re-search with such additional terms one or more times (the query becoming longer and longer), and we call this an *interaction sequence*. Alternatively, at any time a user can start again with a completely new query. When a person starts a browser and then uses UKSearch they are in a *session* which normally lasts until either the browser is stopped or 30 minutes have elapsed. We thus see that the log consists of a series of sessions and that each session comprises one or more interaction sequences, each of which is one or more queries.

The total number of queries in the log for the above year is 40,006 - very small compared even to MSN - but quite large considering its specificity, and large enough for some interesting preliminary experiments. Moreover, considerably more queries have been collected since then. Because the majority of interaction sequences contain only one query, the number of queries which come first in an interaction sequence is 35,463 and it is these which are used here.

The documents for which UKSearch is indexed comprise all the web pages under the www.essex.ac.uk URL plus any files at essex which can be accessed by hyperlinks from such pages.

3. Manual Log Analysis

3.1 Objectives

The aims of the study were:

- To become familiar with the general content and style of the Essex log;
- To establish whether it was possible to understand the meaning of a query, given that they tend to be very short;
- If so, to categorise the queries according to their content;
- Finally, to extract other useful feature information in the process.

3.2 Method

Following an initial examination it was concluded that in the majority of cases it was possible to guess the meaning of a query in the log (this is impossible to prove of course). To put this another way, very few queries seemed to be incomprehensible or illogical. As we have already marked, the domain is highly specific, focusing as it does on university business. It is also highly structured, both in terms of the activities carried out, and the objects and concepts on which they are performed. Thus, even though we have only a small amount of data in absolute terms, it is all about a comparatively small number of well defined subjects.

```
35527    95091B81DF16D8CFA6E7991A5D737741    Tue
May 01 12:57:14 BST 2007       0    0    0
    outside options   outside options   outside
options
35528    95091B81DF16D8CFA6E7991A5D737741    Tue
May 01 12:57:36 BST 2007       1    0    0
    outside options art history    outside
options  outside options art history    outside
options art history
35529    95091B81DF16D8CFA6E7991A5D737741    Tue
May 01 12:57:57 BST 2007       2    0    0
    history art  outside options outside options
art history  history art   history of art
35530    95091B81DF16D8CFA6E7991A5D737741    Tue
May 01 13:01:08 BST 2007       0    0    0
    aa201    aa201    aa201
35531    95091B81DF16D8CFA6E7991A5D737741    Tue
May 01 13:01:32 BST 2007       0    0    0
    aa201 2 au   aa201 2 au   AA201-2-AU
35532    6C50B445B25B4FE374779E054E334292    Tue
May 01 13:08:20 BST 2007       0    0    0    ssh
    ssh      ssh
```

Figure 1 : Appearance of the raw log

```
[Tue,May,1,12,57,14,BST,2007]

>>>
*T *Tue  *     outside options
*T *Tue  *USA  outside options art history
*T *Tue  *USA  history of art
<<<

>>>
*T *Tue  *     aa201
<<<

>>>
*T *Tue  *     AA201-2-AU
<<<

-----------------------------------------

[Tue,May,1,13,8,20,BST,2007]

>>>
*T *Tue  *     ssh
<<<
```

Figure 2 : Log divided into sessions and interaction sequences

We thus proceeded with the manual analysis. The first step was to select a subset of the log to be used. This comprised fourteen complete days, seven days falling in the holidays and seven falling in term-time. Each of the groups of seven days comprised one Monday, one Tuesday and so on. Each day was chosen at random from the corpus. A day is defined to be a 24-hour period starting and ending at Midnight. This process resulted in a set of

1,794 queries, 1,162 falling in term and 632 falling in the holidays.
We then created a set of twenty mutually exclusive subject categories (see Table 1), based on our initial examination of the log. For example, the category Academic or Other Unit deals with queries about departments, schools, research centres and administrative offices (e.g. the Registry) within the university. On the other hand, Parking / patrol staff deals with queries about car parks, parking permits, payment machines, clamping and so on. Each query in the log sample was then assigned to exactly one such category. In the case of ambiguity, the most likely category was chosen. Where a query did not fall into any of the twenty categories, it was assigned to the 21st category Other. Naturally, there are other sets of categories in the literature; for example Spink et al. (2002) have eleven non-exclusive subject categories and Anick (2003) also has eleven categories relating not to topic but to query refinement, and there are various others in the literature. However, due to our domain, it was clearly most fruitful to use our own set.
At the same time, five non-exclusive categories were devised, and zero or more of these were assigned to each query as appropriate (see Table 2). As can be seen in the table, these all deal with capitalisation within the query except one which is concerned with typographical or spelling mistakes.

3.3 Results

The number of queries falling into each semantic category can be seen in Table 3. The six most frequent categories in decreasing order of frequency are Academic or other unit (13%), Computer use (13%), Administration of studies (11%), Person name (10%), Structure and regulations (8%) and Calendar / timetable (7%). These account for 62% of the queries and indeed the top four account for 47%.
What these results tell us is that our queries are concerned with a relatively small number of categories. 93% fall into our twenty classes, with only the remaining 7% in category Other. It can also be seen from the table that the proportion falling into a category falls steeply as we descend the table (ordered by decreasing frequency). In other words, the early categories contain the lion's share of the queries.
We can make a further observation from this: We should tailor our search engine to provide particularly good search or appropriate suggestion or help for the top categories; to be more specific, our engine should know all about academic units, computer software and tools, administration (e.g., student regisration etc) and persons and their exact roles in the university. Conversely, even if we disappoint every person enquiring about a committee, it will not affect our overall success rate very much.
Turning now to the non-exclusive typographical queries (see Table 4), 6% contain a typographic or spelling error. Students very frequently have need of phrases such as plagiarism (e.g. how to avoid being accused of it) and extenuating circumstances (how to get off it when you *are* accused of it and in the mean time they are in need of accommodation, but they are extremely bad at spelling these words. It follows from this that good spelling correction is essential to a high performance engine in this domain.

Query Category	**Examples**
Academic or other unit	`data archive, personnel office`
Computer use	`web mail, printing credit`
Administration of studies	`registration, Tuition Fees`
Person name	`Tony Lupowsky, udo`
Other	`second hand bicycle, theatre props`
Structure & regulations	`corporate plan, dean role of`
Calendar / timetable	`TIMETABLES, term dates`
Map / campus / room	`map of teaching room, 4s.2.2`
Help with studies	`key skills online, st5udent support`
Subject field	`art history, IELTS`
Employment / payscales	`annual leave, staff pay structures`
Course code or title	`cs101, LA240`
Accommodation	`The Accommodation Handbook, accom`
Society	`essex chior, sailing club`
Research	`RPF forms, writing grant proposals`
Degree	`BSc or MA, course catalogue, MA by Dissertation`
Parking / patrol staff	`parking during exams, staff car parking`
Telephone / directory	`internal telephone directory, nightline`
Organisation name	`audio engineering society, ECPR`
Sport	`Cheerleading, fencing`
Committee	`VAG, Ethics Committee`

Table 1 : Topics used in manual classification - typos are original!

Query Category	**Examples**
Acronym lower case	`ecdis, icaew`
Initial capitals	`Remuneration Committee, Insearch essex`
All capitals	`ECDL, CHEP`
Typographic or spelling error	`extenuatin lateness form, pringting credits`

Table 2: Features used in manual classification

Secondly, 19% of queries contain initial capitals (e.g. 'Insearch essex') or all capitals (e.g. 'ECDL'). Thus a search engine should not ignore capitals as most tend to do. Moreover, we need to know what these capitalised terms signify (see later). Thirdly, 3% of queries contain an acronym in lower case (e.g. 'ecdis' which stands for English Classes for Dependants of International Students and Staff - it should be ecdiss of course but in fact it is not). We need to be able to recognise such acronyms and process them accordingly in our search.
Another result arising from this study was that there were a lot of NEs being referred to in the queries (see Table 5).

Some of these are familiar from MUC, TREC etc, for example person names ('udo kruschwitz') and monetary amounts ('£10'). However, others are specific to this domain, i.e. SNEs. Examples include post/role names ('vice-chancellor', 'registrar'[1]), room numbers ('1NW.4.18', '4.305') and online services ('My Essex' - a portal providing information specific to each logged-on student). The table shows 35 SNES in total and this is by no means an exhaustive list.
In the next section, therefore, we move on to a study intended to recognise instances of the SNEs. The next section after that covers possible uses of the SNEs once identified in queries.

Query Category	Frequency	Percent
Academic or other unit	236	13.15%
Computer use	235	13.10%
Administration of studies	198	11.04%
Person name	171	9.53%
Other	161	8.97%
Structure & regulations	145	8.08%
Calendar / timetable	124	6.91%
Map / campus / room	100	5.57%
Help with studies	64	3.57%
Subject field	62	3.46%
Employment / payscales	49	2.73%
Course code or title	44	2.45%
Accommodation	37	2.06%
Society	37	2.06%
Research	30	1.67%
Degree e.g. BSc or MA	27	1.51%
Parking / patrol staff	27	1.51%
Telephone / directory	16	0.89%
Organisation name	13	0.72%
Sport	12	0.67%
Committee	6	0.33%
Query Category	Frequency	Percent

Table 3: Topic analysis of 14-day subset

Query Feature	Frequency	Percent
Acronym lower case	56	3.12%
Initial capitals	207	11.54%
All capitals	140	7.80%
Initial caps or all caps	347	19.34%
Typo or spelling error	111	6.19%

Table 4: Typo / Spelling analysis of 14-day subset

4. Automatic Analysis

4.1 Objectives

The aims of this second study were

- To try to recognise the previously identified SNE types in the query log;
- In so doing to estimate the frequency of these SNEs in submitted queries.

SNE Category	Examples
Person names	richard, udo kruschwitz
Monetary amounts	£10
Post/role names	vice-chancellor, registrar
Email addresses	udo@essex.ac.uk, udo (!)
Telephone numbers	+44 (0)1206 87-1234
Room numbers	1NW.4.18, F1.26, 4.305, 5A.101, 4SA.6.1, 4B.531
Room names	Senate Room
Lecture theatres	Ivor Crewe Lecture Hall
Buildings	Networks Building, sport centre
Depts / schools /units	Department of Biological Sciences, International Academy, Estates, Nightline, chaplaincy, data archive, Students' Union
Campuses	Colchester, Southend, East-15 Loughton campus
Shop names	Hart Health & Beauty, wh smith
Restaurants / bars	blue cafe, sub zero
Research centres	Centre for Environment and Society, Chimera
Research groups	LAC
Degree codes	RR9F, g4n1, GC11, m100
Degree names	FINANCIAL ECONOMICS, MA TESOL, mpem
Course codes	HS836, PA208-3-AU sc203
Course names	Machine Learning, toefl
Online services	DNS, subscription lists, ePortfolio, MyLife
Software	spss14, MATLAB, limewire
Seminar series names	spirit of enterprise, FIRSTSTEPS
Buses	X22, 78
Banks	lloyds, barclays bank
Addresses	rayleigh essex SS6 7QB
Accommodation	FRINTON COURT, sainty quay
Societies	ORAL HISTORY SOCIETY, metal society, CHRISTIAN UNION
Projects	tempus
Publications	wyvern
Documentation	Access Guide, ug prospectus, student handbook, quality manual
Regulations & policies	asset disposal policy, regulation 4.32, higher degree regulations
Relevant parliament acts	data protection
Events	Freshers Fair, comedy nights
Forms	ALF form, p45, password change form
Equipment	scanner, blackberry, defibrillator

Table 5: Examples of 35 SNEs used for training

[1] The Vice-Chancellor despite the name is in fact the head of an English university. A typical eccentricity. The Registrar is the most senior officer in the university concerned with the administrative side, e.g. student records, finance, all administrative staff etc).

Named Entity	TE	C	M	F	A	P	R
accommodation	6	0	0	0	280	0	0
addresses	1	0	0	0	280	0	0
banks	4	0	0	0	280	0	0
buildings	20	2	3	0	275	1	0.4
buses	4	0	0	0	280	0	0
campuses	6	2	1	0	277	1	0.67
course_codes	31	1	8	1	270	0.5	0.11
course_names	6	0	1	0	279	0	0
degree_codes	22	0	0	0	280	0	0
degree_names	32	0	6	0	274	0	0
depts_schools_units	107	15	5	1	259	0.94	0.75
documentation	21	5	17	45	213	0.1	0.23
email_addresses	4	0	0	1	279	0	0
equipment	13	0	0	3	277	0	0
events	2	0	1	0	279	0	0
forms	29	1	2	0	277	1	0.33
lecture_theatres	3	0	0	0	280	0	0
monetary_amounts	0	0	0	0	280	0	0
online_services	66	41	5	0	234	1	0.89
person_names	338	3	10	0	267	1	0.23
post_role_names	72	1	1	1	277	0.5	0.5
projects	1	0	0	4	276	0	0
publications	1	0	2	0	278	0	0
regs_and_policies	54	2	2	0	276	1	0.5
rel_parliament_acts	1	0	3	0	277	0	0
research_centres	16	0	0	0	280	0	0
research_groups	4	1	0	0	279	1	1
restaurants_bars	13	4	0	7	269	0.36	1
room_names	52	11	5	0	264	1	0.69
room_numbers	39	0	0	0	280	0	0
seminar_series	3	0	0	0	280	0	0
shop_names	9	0	0	0	280	0	0
societies	11	0	0	0	280	0	0
software	37	1	2	0	277	1	0.33
telephone_numbers	7	0	0	0	280	0	0
All NEs	1035	90	74	63	9293	0.59	0.55

Table 6: Results of Training. TE is the number of training examples used. C, M, F & A are the numbers Correct, Missed, False positive and Absent. P & R are Precision and Recall defined as follows: P=C/(C+F); R=C/C+M)

4.2 Method

The supervised learning algorithm chosen was the Maximum Entropy text classifier forming part of the OpenNLP (2010) suite. This was selected on the basis of its reputation in this kind of task, its ready availability and its integration into the Sheffield GATE system.

An initial sample of 1,035 training instances of the 35 SNEs in Table 5 were identified in the log. As can be seen in the table, the number of instances used was not the same for each SNE. Each SNE was then searched for at the Essex web site and up to ten documents containing it were identified (sometimes there were none). From each document, one or more snippets centred on the SNE were extracted (often there was only one). Each snippet had up to five tokens on each side. A token was deemed to be anything separated by one or more white space characters on each side. In cases where the SNE was for example very close to the start of a file, the context could be less than five tokens. Punctuation characters such as '.' and ',' remained attached to a word except in the case where they were originally attached to the end of the NE itself. In this case, they were preceded by a space.

For testing data, the first 500 queries in the complete log were extracted. Of course, each one could either contain an SNE or not. An attempt was made to find a snippet for each query by the means just described for the training instances. As the search was exact and queries were not by any means guaranteed to occur even in one document, there were instances of queries for which no snippet could be produced. For a query in test set, a maximum of one snippet was extracted. This contrasts with the training set where there could be ten or more.

For training, there were two scenarios in which we were interested:

- Train with snippets; test with snippets.
- Train without snippets; test without snippets.

Here we report on Scenario One. We have a set of 35 SNEs and for each set we have one or more training examples for each SNE (see Table 6, column 2 for the numbers of instances). Each such training example comprises the SNE in the middle with five tokens on each side.

We next train a classifier for the first SNE, e.g. accommodation, by presenting each of the training examples. We then proceed to the second SNE, e.g. addresses and build a separate classifier for that, and so on.

For testing, we take each test query, complete with the snippet we have extracted for it. We present it to the first classifier (accommodation), whereupon the classifier decides whether it is an instance of the SNE accommodation or not. We do the same with the other queries. We then present them to the second classifier (addresses) and so on. Notice therefore that this is not a mutually exclusive classification.

In Scenario One we have snippets for all SNE instances whether for training or testing. This is the most powerful case but we wanted to see whether we could do without the snippets. For one thing, the snippets may not occur in the documents. For another, they may be false in some way, caused by an accidental match - e.g. Athens could be a place in Greece or a software system in the library - most likely our log instances are the latter not the former. Another problem is that we have to find snippets for all our testing instances and it is not really practical to check all these by hand.

We therefore wanted to see whether we could do without snippets and this is the rationale behind Scenario Two above, though we have not investigated this yet.

4.3 Results and Discussion

The results of the experiment in recognising SNEs using Scenario One are shown in Table 6. We will now explain the columns. Training Examples (TE) is the number of examples which were had for each SNE. Thus for person names we had 338 typographically distinct examples. Mostly these were completely different names, but sometimes there were one or two variants of a name (e.g. ‘udo’, ‘Udo’ etc).

C(orrect) is the number of cases where the whole query was an SNE, and the system recognised that SNE correctly and demarcated it exactly. It is our goal that all

classifications should be C.
If the query was an SNE and the system did not recognise it, then the classification was M(issed). This includes the case where it did not recognise it at all, and the case where it recognised part of it correctly. So if we had 'Ivor Crewe Lecture Theatre' as the query and we did not recognise it at all as a lecture theatre, that would be M. However, if we recognised 'Ivor Crewe' as the lecture theatre, we considered that M also. (This might be ineXact in a TREC type analysis, but we are not using lenient measures here.)
Next, if the query was not an SNE at all but the system recognised it as one, this was F(alse positive). Finally, if the query was not an SNE and the system did not recognise it as one, this was A(absent).
Precision and Recall are defined in this study as follows. P=C/(C+F). R=C/(C+M).
The column TE shows the breakdown of SNEs found in the training data. The breakdown in the test data is the sum of columns C and M.
In general, the results show that if an SNE is clearly defined and we have a good number of training examples for it, we can expect a good performance. Precision was 1 for buildings, campuses, forms, online services, person names, regulations and policies, research groups, room names and software. P was 0.94 for departments / schools / units. The most interesting of these are departments / schools / units, online services and room names for which there were 15, 41 and 11 correct instances respectively. The numbers for the other SNEs whose P was 1 were much smaller.
About false positives, F, we can see that the algorithm is highly disinclined to make them, meaning that precision is very good. Compare column F(alse positive) with column A(bsent). The documentation SNE was the only one where things went badly wrong. We explain this next.
Conversely, if an SNE is not cleary defined or we do not have a lot of examples, we can only reasonably expect a poor performance. A good example of unclear definition is the SNE documentation. There are 21 training examples (the same as for buildings where P was 1) but P=0.1 because there are 45 false positives. The algorithm does not know what to recognise because we have not made it clear. We have not made it clear because we ourselves are not clear.
The context we use is also very important. Our algorithm was simplistic because it did not distinguish between different types of documents. On the one hand we have HTML and on the other PDF, PS, DOC, XLS etc. Each of these has different characteristics. Moreover, in a particular document type there are different types of text. In a running text, our SNE occurs in a 'naturalistic' context which makes sense. This is optimal for the algorithm because the SNE is syntagmatically (and of course semantically) linked to the words around it. As is well-known, a grammatical context contains patterns of occurrence which are what the algorithm is primarily tuned to pick up. Now, let us contrast this with the other types of document. There are two which we would like to mention. The first is the table, in which our SNE is commonly found. The 'neighbours' of the SNE are adjacent columns in the table and of course these may not be related at all to the SNE. If they are, this relation will not be syntagmatic and hence may undermine the patterns being deduced from the naturalistic training examples. The second type of document is the high level web page. The classic example is www.essex.ac.uk. You will see that this is not text at all. Instead it is a series of 'advertisements' for different and quite unrelated activities in the university. Once the text is extracted from the HTML, the context is often a jumble of unrelated words. Our five-word context on each side can sometimes contain extracts from two or even three completely different 'advertisements'. Naturally, this is not conducive to learning. So in conclusion, we need to be much cleverer about how we select contexts for both training and testing examples.
Another very interesting finding from this study concerns SNEs which are identified by their internal 'syntax' as well as by their use. A good example is room numbers (see Table 5, row 6). As can be seen, these follow arcane laws worked out by the administrators at Essex in the 1960s. After a few years at the university, a person can often predict the location of the room because it often includes the square at which you enter it, the cardinal point of the square and the floor. So, 1NW.4.18 means Square 1, North West corner, floor 4, room 18. Our observation in experiments was that the algorithm was remarkably good at recognising these names. Unfortunately, however, none occurred in the test data we are working on here, so this result is not reflected in the Table 6. We should also remark that the features used and the parameter settings for the learning algorithm for this study were the standard settings.
Turning to the number of training examples used in the study, this was chosen to be around 1,000 in total. As can be seen in Table 6, column 2, the number of examples varies from 0 up to 338. Clearly, there should be a large number of examples for each SNE and probably the number should be the same across SNEs.
The number of testing instances was limited to 500 for practical reasons and this is obviously much to small, especially as it was only possible to find contexts for 280 of these. The C and M columns added up tell us how many instances of an SNE there were in the test data. The maximum is online services at 46. For most SNEs there were only a handful. The proportion of queries containing an SNE at all can be estimated (as an upper bound) as the sum of C and M for the bottom row which covers all SNEs, divided by the number of training instances, 90+74/280, i.e. 59%. This is high, but so is the number of SNEs we are working with.
Concerning text contexts (up to five tokens either side), we used just one such context for determining whether a complete query was an SNE. Recall that for training we used up to ten. Given that our contexts are less than perfect, it might have been better to use several for testing and to pool the results in some way. More example contexts should probaably used for training also, where they are available.
Finally, we insisted in this study that an entire query was an SNE, for example 'udo kruschwitz' . In other words a query which contained an SNE plus other information (e.g. 'udo kruschwitz conference') was not used in this experiment. Such queries are not that numerous, but we would like to work out a way of dealing with them. One possibility is to submit various subsets of the query to see if they are SNES. For example, we could submit 'udo kruschwitz' and 'kruschwitz conference', leading to the discovery that the former was an SNE while the latter was not.

5. Using the SNE for Effective QA

We have talked about recognising SNES. We will now outline some possible applications for effective searching / QA within our domain.

Firstly, a search for an SNE such as a person should match all references to a person. Some references include the name or some substring of it. So, Kruschwitz should match Udo Kruschwitz. References to one part of a name should also match the other where it is known. So Udo should match Kruschwitz and vice versa. Next, a name should match other forms of co-reference to it, i.e. Kruschwitz should match 'he' in appropriate contexts. Finally, a name should match a post where it is held by the person. Thus Kruschwitz should match Director of Recruitment and Publicity.

Next, SNEs do not exist in isolation. Rather, they are linked. For example course codes are linked to course names, degree codes are linked to degree names, and courses are linked to degrees. It follows that if someone enters a course code we want to return the course name and very possibly the appropriate degree names/codes as well.

Similar remarks can be made about departments, research centres and research groups. For example, centres and groups tend to be linked to departments, so someone asking about a group may be interested in other work in the parent department. Person of course are associated with groups, centres and departments. So a search about a person could require information about the department which that person belongs to.

Finally, for room numbers they should be associated with the appropriate building, with the department in question and possibly in the case of the room number of an office with the name of the occupant.

In summary, then, the number of links and associations between SNEs in the university domain is great as is their potential if properly exploited within a search engine or QA system.

6. Conclusions

In this article we started off by describing the setup at Essex together with the query log and UKSearch engine. We then outlined an initial manual study in which a selection of queries entered into UKSearch over fourteen days distributed throughout the year were classified using twenty subject categories plus a default 'Other'. We also asigned typographical features. Next we introduced a set of 35 SNEs - domain specific NEs which were occurring regularly in the log. We outlined an experiment in which we attempted to recognise these automatically in the log using a Maximum Entropy tagger together with snippets extracted automaticaly from the Essex website. Finally, we explained why we are intersted in SNEs in the first place and we elaborated on specific applications to which both SNEs and - most importantly - domain-specific relations *between* SNEs can be put.

7. References

Anick, P. (2003). Using Terminological Feedback for Web Search Refinement - A Log-based study. Proceedings of SIGIR, July 28 - August 1, 2003, Toronto, Canada, 88-95.

CLEF (2010). http://www.clef-campaign.org. Accessed 2010.

Grishman, R., Sundheim, B. (1996). Message Understanding Conference - 6 : A Brief History. www.aclweb.org/anthology/C/C96/C96-1079.pdf

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

Jansen, B. J., Spink, A., Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the web. *Information Processing & Management*. 36(2), pp. 207-227.

Kruschwitz, U. (2003). Automatically Acquired Domain Knowledge for ad hoc Search: Evaluation Results. In *Proceedings of the 2003 Int. Conference on Natural Language Processing and Knowledge Engineering, Beijing*, 525-532.

Kruschwitz, U. (2005). *Intelligent Document Retrieval: Exploiting Markup Structure*. New York, NY: Springer.

Kruschwitz, U., Al-Bakour, H. (2005). Users Want More Sophisticated Search Assistants - Results of a Task-Based Evaluation. *JASIST*, 56(13), 1377-1393.

Kruschwitz, U., Webb, N., Sutcliffe, R. F. E. (2009). Query Log Analysis for Adaptive Dialogue-Driven Search. In J. Jansen, I. Taksa and A. Spink (Eds.) *Handbook of Web Log Analysis* (pp. 389-414). Hershey, PA: IGI Global.

NTCIR (2010). http://research.nii.ac.jp/ntcir/. Accessed 2010.

OpenNLP (2010). http://opennlp.sourceforge.net/. Accessed 2010.

TREC (2010). http://trec.nist.gov/. Accessed 2010.

Log-Based Evaluation Resources for Question Answering

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

Abstract

In this paper we discuss the applicability of evaluation resources for the development and improvement of question answering systems. The logfiles of queries and user actions which have been available at the Cross Language Evaluation Forum (CLEF) are discussed as primary examples for log-based evaluation resources for information retrieval. Some evidence is given that these logs do contain entries which could be interesting for question answering research. It is also discussed that current question answering systems might need to adapt a more robust style in order to serve information needs expressed in many of these queries.

1. Information Retrieval Logs and Question Answering

Information retrieval and question answering systems are typically considered related by separated research areas. Question answering analyzes natural language sentences and returns only the (often brief) answer to a question. On the other hand, information retrieval is based on the "bag of words" paradigm which does not analyze the query in a syntactical way and basically treats each word differently.

Users are not always aware that such different systems exist. The great success of web search engines which work like information retrieval systems has shown that the short query is a preferred way of asking for information. Question answering systems which require a longer query in the form of a well formed natural language question are commercially much less successful than web search engines, for example.

Nevertheless, users expect information retrieval systems to "understand" natural language to a certain extent. Often, query terms are combined to phrases ("jornais of Leira") and sometimes even complete sentences are entered. Obviously, users tend not to worry about differences in the functionalities of systems but expect a sort of a hybrid information access. Typically, a system should be able to handle very brief input. On the other hand, the system should also be able to deal with natural language when the user makes the effort to enter longer phrases or even full sentences.

Consequently, a system should be able to act differently according to the query entered. This is in line with much information retrieval research which demands query specific treatment (Mandl & Womser-Hacker 2005). Many search engines have also put this demand in practice for some specific questions. If a search engine detects that a query is better answered by a specific result it may only or primarily display that result. Examples are addresses which are answered with maps, names of soccer clubs which are answered with an overview of recent and upcoming games and city names which are answered with train line information. We propose that a search service should also switch between a "bag of words" approach and a question answering approach based on the input.

The remainder of the article will discuss the availability of evaluation resources for such a switching mode. Due to the great success of information retrieval systems especially in web search engines, many evaluation resources in the form of log files of web searchers have been created. However, only very few of them are available for interested researchers. So far, three logfiles which have been used in comparative evaluation campaigns are known. An overview is given in table 1 and the resources are described in the following section.

Year	Origin	Size	Type
2007	MSN	800.000 queries	Query log
2009	Tumba!	350.000 queries	Query log
2009	TEL	1.870.000 records	Query and activity log
2010 (planned)	TEL	extended	Query and activity log
2010 (planned)	DIPF.de	-	Query log

Table 1: Logfile resources at CLEF.

2. Information Retrieval Evaluation Resources

The Cross Language Evaluation Forum[1] (CLEF) is a large European evaluation initiative dedicated to cross-language retrieval for European languages. CLEF was implemented as a consequence to the rising need for cross- and multi-lingual retrieval research and applications. CLEF provides a multi-lingual testbed for retrieval experiments. The evaluation campaign of CLEF comprises several components: the evaluation methodology, the evaluation software packages, the data collections, the topics, the overall results of the

[1] http://www.clef-campaign.org

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;-;;;

Table 2: Example records from the TEL log.

participants, the assessed results of the participants, and the calculated statistical results.

LogCLEF[2] was the first track at an evaluation campaign dedicated to providing logfiles and fostering the analysis of user behaviour based on logs. The goal is the analysis and classification of queries in order to understand search behavior in multilingual contexts and ultimately to improve search systems.

The first task with logfiles was organized within the track GeoCLEF 2007 which investigated and provided evaluation resources for geographic information retrieval (Mandl et al. 2008). The query identification task was based on a query set from MSN. The goal was the identification of queries with a geographic component. Three types of queries were defined "Yellow page", "Map" and "Information" (Li et al. 2008). According to a preliminary analysis of the complete set, some 400.000 entires (some 50%) of the queries contained geographic terms (Li et al. 2008). The data set is no longer distributed by Microsoft.

For LogCLEF 2009, two data sets were provided. One data set used at CLEF 2009 consists of the 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 type of these action logs an example of the possible usage of the portal is described in the following. 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. After the search is initiated the result page appears, where results are ordered by collections and the results of the top collection in the list are presented with brief descriptions.

All these type of actions are logged and stored by TEL in a relational table, where a table 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 2.

The second data set for LogCLEF 2009 is a web search engine query log from the Tumba! search engine.

LogCLEF 2010 attracted 13 participating groups from nine countries. They developed or modified their systems and ran experiments with the data. The detailed results for all sub tasks are provided in the overview paper (Mandl et al., 2010).

3. Question Style Queries in Query Logs

In this section, we give some evidence of queries which could be interesting for question answering systems or for switching systems as outlined in the introduction section.

82nd airborne who fought in Vietnam
health effects of arsenic exposure to a patient who is HIV positive
when is easter in 2007
when is labor day
when is spring break in California
when is hurricane season in florida
gastric bypass surgery in illinois who accept medicaid insurance
who are the men that paved the weight to a development of the biology of a science
who took control of egypt in 306 bc

Table 3: Examples for queries from the MSN query logfile.

As any web search log file, the MSN log used in GeoCLEF 2007 contains mainly brief queries but also some longer ones. We browsed the longer queries and

[2] http://www.uni-hildesheim.de/logclef

found many examples which have the form of natural language questions and which are suitable for question answering system. Some examples are shown in table 3. A systematic way to extract these queries has not been identified yet. It could be an option to test if the queries can be mapped to a syntactic structure by a parser.

Within the logfile from the TEL service (The European Library) most longer queries seem to contain book titles which is reasonable considering the content of the service. We managed to identify a non significant number of potentially question style queries by exploring the data set which is shown in table 4.

a biographical register of the university of oxford ad1501 1540 by ab emden
what was the story where a princess was locked in a tower gaurded by a beast
what happens to elements and compounds in salt when put into water
tratados internacionales en materia de propiedad intelectual
who was the first known european to view america
alphabetical list of persons who polled for a member of parliament to represent
who created the first library
images of what happens when plants get no light
biographies of people who succeeded after several failures
when was the last major impact of a NEO
where can i find phone books street directories tax and church records for etterbeek belgium
psychosociaal impact of computers on seniors

Table 4: Examples for queries from the TEL logfile.

Ghorab et al. 2009 found that over one quarter of all reformulations in the TEL are additions or deletions of stop words. Also question words like “where” or “when” are common stop words in information retrieval systems. Prepositions are typical in the reformulation set, too. This means that users are not aware of the functionality of the system because stop words are ignored by information retrieval systems and also by the TEL digital library.

The use of prepositions hints at users thinking that the system has natural language functions. In such a case, some sort of natural language processing (NLP) could be initiated. Obviously, this would require NLP at the indexing phase. Even if no NLP functions are available, the system could internally search with phrases (phrase of “PREP NOUN”) and merge the result with a regular search. We also observed the frequent use of prepositions in the Tumba! search engine log. In the MSN log, prepositions belong to the most frequent terms.

4. Outlook

CLEF has created evaluation resources for logfile analysis which can be used for comparative system evaluation. The available files do contain queries which could be interesting for question answering systems. They contain full sentences as questions or phrases which cannot be processed appropriately by the “bag of words” approach.

5. References

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: Peters, C.; Di Nunzio, G.; Kurimo, M.; Mandl, T.; Mostefa, D.; Peñas, A.; Roda, G. (Eds.): *Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments: Proceedings 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece. Revised Selected Papers.* Berlin et al.: Springer [Lecture Notes in Computer Science] to appear. Preprint in Working Notes: http://www.clef-campaign.org/2009/working_notes/

Li, Z., Wang, C., Xie, X., Ma, W.-Y. (2008). Query Parsing Task for GeoCLEF2007 Report. In: *Working Notes 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007*, Budapest, Hungary, http://www.clef-campaign.org/2007 /working_notes/LI_OverviewCLEF2007.pdf

Mandl, T., Gey, F., Di Nunzio, G., Ferro, N., Larson, R., Sanderson, M., Santos, D., Womser-Hacker, C., Xing, X. (2008). GeoCLEF 2007: the CLEF 2007 Cross-Language Geographic Information Retrieval Track Overview. In: Peters, C.; Jijkoun, V.; Mandl, T.; Müller, H.; Oard, D.; Peñas, A.; Petras, V.; Santos, D. (Eds.): *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum. CLEF 2007, Budapest, Hungary, Revised Selected Papers*. Berlin et al.: Springer [Lecture Notes in Computer Science 5152] pp. 745--772.

Mandl, T., Womser-Hacker, C. (2005). The Effect of Named Entities on Effectiveness in Cross-Language Information Retrieval Evaluation. In: *Proceedings of 2005 ACM SAC Symposium on Applied Computing* (SAC). Santa Fe, New Mexico, USA. March 13.-17. pp. 1059--1064.

Mandl, T.; Agosti, M.; Di Nunzio, G.; Yeh, A., Mani, I.; Doran, C.; Schulz, J.M. (2010): 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.): *Multilingual Information Access Evaluation Vol. I Text Retrieval Experiments: Proceedings 10th Workshop of the Cross-Language Evaluation Forum, CLEF 2009, Corfu, Greece. Revised Selected Papers.* Berlin et al.: Springer [Lecture Notes in Computer Science] to appear. Preprint in Working Notes: http://www.clef-campaign.org/2009/working_notes/LogCLEF-2009-Overview-Working-Notes-2009-09-14.pdf