ADCS 2006

Proceedings of the Eleventh Australasian Document Computing Symposium,

11 December 2006

Edited by
Peter Bruza, Amanda Spink and Ross Wilkinson

Proceedings of the Eleventh Australasian Document Computing Symposium,
held at the Queensland University of Technology
11 December 2006.

Published by the
Faculty of Information Technology,
Queensland University of Technology,
Australia.

Editors:

Peter Bruza
Amanda Spink
Ross Wilkinson

ISBN 0-XXXXXXX-X-X

www.adcs06.fit.qut.edu.au/

ADCS 2006
The Eleventh Australasian Document Computing Symposium

Brisbane, Australia
11 December 2006

Chairs' Preface

These proceedings contain the papers of the Eleventh Australasian Document Computing Symposium hosted by, and held within, the Faculty of Information Technology at the Queensland University of Technology.

The keynote address, twelve papers and two posters reflect the breadth and interest of the Australian research community in the area of document computing.

The twelve papers here were selected from seventeen submissions. Every paper had three peer reviews. Submitted papers were anonymously reviewed at their full length by experts in the area. Dual submissions were explicitly prohibited.

The members of the program committee and the extra reviewers deserve special thanks for their contribution to ADCS2005.

The symposium includes many formal presentations, but perhaps its greatest benefit lies in the opportunity it provides for document computing practitioners to get together informally and to share ideas and enthusiasm.

Peter Bruza
Amanda Spink
Ross Wilkinson

Symposium Chair

Peter Bruza	Queensland University of Technology

Program Co-chairs

Amanda Spink	Queensland University of Technology
Ross Wilkinson	CSIRO

Program Committee

Vo Ahn	University of Melbourne
Peter Bailey	CSIRO
Wray Buntine	University of Helsinki
Lawrence Cavedon	RMIT University
Robert Dale	Macquarie University
Peter Eklund	University of Wollongong
Tomas Gedeon	Australian National University
Schlomo Geva	Queensland University of Technology
Baden Hughes	The University of Melbourne
Judy Kay	University of Sydney
Mun Kew Leong	Institute for Infocomm Research
Chris Lueg	University of Tasmania
Rob McArthur	CSIRO
Gitesh K. Raikundalia	Victoria University
Andrew Trotman	University of Otago
Andrew Turpin	RMIT University
Anne-Marie Vercoustre	INRIA, France

ADCS Advisory Committee

Peter Bruza	Queensland University of Technology
David Hawking	CSIRO
Judy Kay	University of Sydney
Alistair Moffat	The University of Melbourne
Ross Wilkinson	CSIRO
Justin Zobel	RMIT University

Contents

Some Observations on User Search Behavior

Yuye Zhang
NICTA Victoria Research Laboratory,
Department of Computer Science and Software Engineering
The University of Melbourne, Australia
zhangy@csse.unimelb.edu.au

Alistair Moffat
Department of Computer Science and Software Engineering
The University of Melbourne, Australia
alistair@csse.unimelb.edu.au

Abstract *We explore some issues that arise in the way that users interact with a web search engine, as evidenced by the records of their interaction provided by query and clickthrough log data. Our observations are derived from approximately fifteen million user queries recorded by the* `search.msn.com` *search service in May 2006.*

Keywords Log analysis, user behavior, search.

1 Introduction

Query logs from large scale search engines have always been a research commodity, providing crucial insights into the interaction between the users of the system and the system itself. The results of studies on search engine query logs can be applied to a range of fields in computing, including contributions to the fields of user interface design, to search result reranking, and to predictive caching and prefetching [Fagni et al., 2006].

Although there has been much done in analysis of logs containing queries submitted to a range of search engines [Silverstein et al., 1999, Spink et al., 2002, 2001, Lempel and Moran, 2003], research into the usefulness of clickthrough data as a model of user search behavior has received little attention due to a lack of public datasets. Because clickthroughs are indicative of a user's preference with respect to a particular query, they can be used as evidence to explore, for example, web personalization [Eirinaki and Vazirgiannis, 2003], and implicit relevance feedback [Joachims et al., 2005, Joachims, 2002, White et al., 2005].

In this paper we report findings from our analysis of a recently released log for the Microsoft MSN Search whole-of-web search engine (`http://search.msn.com`) containing approximately fifteen million queries and the corresponding clickthrough data, both of which are representative of a one month period in May 2006. We conduct our analysis by reporting key statistics of the dataset, and provide detailed insight into three major aspects of this query log: queries, sessions, and clickthroughs.

The dataset examined in our study is both large and recent, and is accompanied by clickthrough outcomes. The trends we have extracted from this dataset are thus both topical and timely, and provide evidence of a range of trends in user search behavior.

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.

2 Definitions

To ensure consistency of terminology, we make use of these straight-forward definitions:

Query: A string issued by an user to a search engine as a request for information.

Term: Individual words within a query, separated by whitespace. Terms may include alphanumeric characters, punctuation and other symbols. The number of terms in a query is the query *length*. Note that multi-word quoted phrases are considered multiple terms.

Session: A set of queries from a particular user, deemed (usually by a heuristic) to be part of a single interaction with a search engine. The session might include queries that relate to more than one information need, or topic. The *length* of a session is the number of queries contained in it.

Results Page: An ordered list of results presented to the user for a given query. The results page usually contains links to ten Results, plus a variable number of sponsored and other commercial links.

Result: An individual URL on the Results Page (plus a snippet of representative text extracted from that page), providing access to a document suggested by the search system as being an answer to the query.

Clickthrough: The action of the user in clicking on one of the Results listed in a Results Page, in order to access the page at the indicated URL.

Using these definitions we can see that a session contains one or more queries, each composed of one or

more terms. Each query execution generates a Results Page, and as a result of examining that Results Page, the user may generate zero, one, or multiple clickthroughs.

3 The MSN dataset

The MSN Search dataset was released as part of the "Microsoft Live Labs: Accelerating Search in Academic Research" incentive in 2006 (`http://research.microsoft.com/ur/us/fundingopps/RFPs/Search_2006_RFP.aspx`). This dataset contains approximately fifteen million queries originating from users from the United States during May 2006, as recorded by the MSN search engine. All queries contained within the dataset are timestamped, sessionized and also anonymized to remove any personally identifiable information. Clickthrough data is provided in a separate file, with the two linked by a query identifier. Each clickthrough record contains that identifier, a timestamp, the URL accessed, and information about the rank of that result in the results page (being positions 1 to 10 on the first results page, 11 to 20 on the second, and so on). No information regarding unclicked results is retained.

In examining the logs, it quickly became apparent that not all of the queries in the query log were issued via the MSN Search frontend, and that the log included queries that originated from other external sources such as Web APIs, toolbars, third party programs, and so on. This presented a concern during preliminary analysis, as closer examination of the hundred largest sessions in the dataset revealed that around 90% of them appeared to be machine-driven. In particular, the largest session in the dataset (containing over $30{,}000$ queries, issued at a consistent rate of around three queries per second) stepped in sequence through a sorted list of URLs to query their backlinks via the `linkdomain:` search option. Additionally, the second and third largest sessions (both consisting of $3{,}081$ queries) contained repeated requests for two different sets of exactly ten queries, again at a rate higher than would normally be associated with a real "user". We can only conclude that these two sessions were the result of someone exploring the search API, possible with a faulty program. Only two queries in these three sessions had any clickthroughs associated with them. Other common session variants of dubious usefulness included sorted alphabetical lists relating to some given niche such as real estate, as well as repeated non-sensical queries.

In one sense, the log thus depicts a "warts and all" approach to querying, in that it fairly reflects the workload that the system is asked to execute (whether erroneously, surreptitiously, or maliciously is unknown). On the other hand, our purpose in this investigation is to estimate the behavior of genuine users, and these sessions significantly distort the underlying trends – in reality it is extremely unlikely that a user would issue more than 100 queries in a single session.

In order to report usage patterns of users rather than general web search engine traffic, we thus faced the issues of:

- Whenever possible, eliminating machine-generated sessions; and
- Ensuring each session represents one exchange between a single user and the search engine.

The latter is difficult to deal with, as it appears that the sessionization heuristic make use of the user's IP address, which can be identical for multiple users on a network behind, for example, a proxy server. Manual segmentation into individual sessions is not an option – the sheer size of the dataset, and the fact that a session can genuinely include multiple query threads (meaning queries can legitimately occur in very short intervals), makes this impossible.

Hence, we chose to filter the query log by removing all queries which did not have any corresponding clickthroughs. We believe that this strategy correctly removes all query requests originating from automated sources, as they are generally concerned with aggregating result page data and not exploring individual results. After applying the filtering process, out of the hundred longest sessions, only nine remained, matching our initial informal exploration.

Unfortunately, the filtering strategy has the disadvantage of also removing any queries originating from real users for which no results were clicked. In these cases non-clicking is valid information, since it suggests that the user was either not interested in any of the proposed results, or that their information need was satisfied by the snippets alone. Since these two cases are impossible to differentiate anyway, we felt that the removal of such sessions was an acceptable compromise in order to be sure that the machine generated traffic had been largely removed.

In the remainder of the paper, results are presented primarily for the filtered query log (note that the filtering was not applicable to the log of clickthrough data), but those results are contrasted with the unfiltered query log where it is appropriate to do so.

4 General statistics

Table 1 provides a range of statistics extracted from the query log (before and after filtering) and the clickthrough log. Over a third of the original queries were removed by the filtering process. Of those queries that had clickthroughs associated with them, around two thirds had a clickthrough to the first of the results, and the average first clickthrough position was 2.2. In the unfiltered dataset, there was fewer than one clickthrough recorded per query, on average; removal of the queries that had no clickthroughs increased this ratio to around 1.4.

Figure 1 shows the distribution of queries across the individual days in May 2006, beginning on Monday,

Attribute	Original	Filtered
Number of queries	14,923,285	8,831,275
Number of unique queries	7,095,622	3,875,436
Number of terms	35,824,851	20,641,810
Number of unique terms	2,605,699	1,151,998
Number of sessions	7,470,913	5,684,599
Average query length (terms)	2.401	2.337
Median query length (terms)	2	2
Average session length (queries)	1.997	1.554
Median session length (queries)	1	1
Average time between queries in a session (mm:ss)	4:21	7:28
Median time between queries in a session (mm:ss)	1:13	3:20
Number of clickthroughs	12,251,067	
Number of clickthroughs at rank 1	6,074,872	
Average clickthroughs per query	0.821	1.387
Median clickthroughs per query	1	1
Average rank of first clickthrough in a query	2.161	
Median rank of first clickthrough in a query	1	
Average time between clickthroughs in a query (mm:ss)	2:03	
Median time between clickthroughs in a query (mm:ss)	0:47	

Table 1: Key statistics describing the query log and clickthrough log, before and after filtering to remove machine-generated queries.

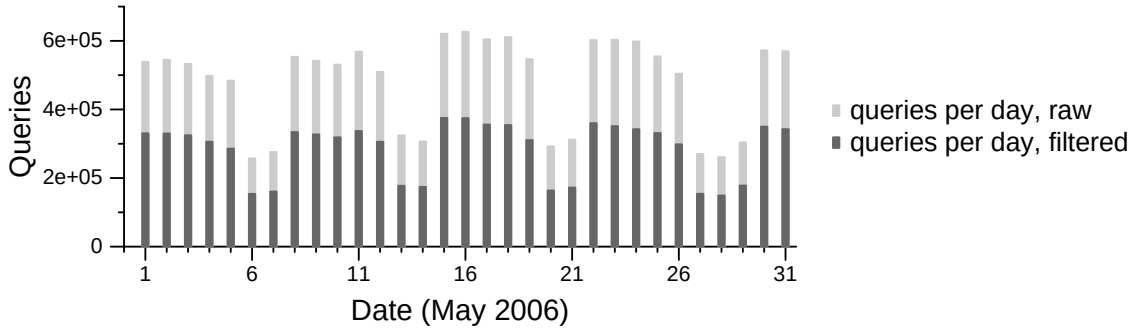


Figure 1: Daily query volumes for the collection period during May 2006, starting from Monday, May 1st. Query activity generally peaks on Monday, with a gradual drop off through until Friday, and a significant reduction through the weekend. Monday 29 May was a national holiday in the United States.

May 1st. It is clear that the volume of queries received by the search engines follows a general pattern of peaking early in the week and dropping steadily until Friday, with a sharp decrease over the weekend. This reflects quite accurately the weekly working cycle of the average white collar worker, and perhaps even captures the trend in which query volumes drops towards the end of the working week indicating a possible decrease in individual productivity. It also suggests that either search activity has become an important and integral part of the standard office routine, or that employees exploit company resources (time and connectivity) to undertake private searches.

Similarly, Figure 2 depicts an hourly breakdown for queries and clickthroughs received by the search engine, amalgamated across the entire month. Note that the logs supplied are a "representative sample" of US-originated queries during this period, but are not comprehensive. That is, the total hourly/daily/monthly search volume handled by the Microsoft engine is unknown, but the pattern of usage depicted in Figures 1 and 2 is accurate.

When broken down across the day, a pattern emerges where query volumes rise substantially from early morning (around 4am PST, at which time it is 7am in New York) peaking at around noon PST, when the whole country is at "work", and then decreasing steadily through until midnight PST. The ratio of clickthroughs to queries – both before filtering and after filtering – is relatively constant. This suggests that the machine-generated sessions that were removed by the filtering step are distributed through the day in the same pattern as are the user-generated query requests.

In both a daily sense and a hourly sense the filtering process does not appear to have affected the trends within the data.

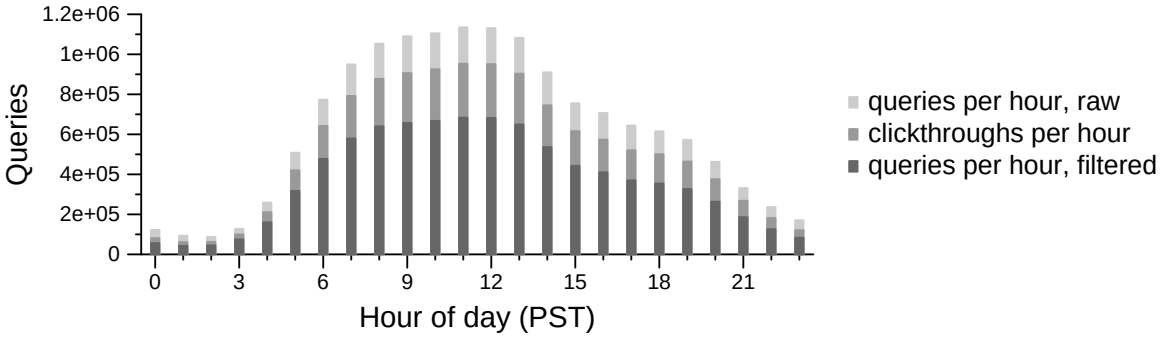


Figure 2: Hourly query and corresponding clickthrough volumes during the collection period. Search activity rises sharply during the early morning (when the US eastern states start work), peaking around noon and gradually drops off in the evening. Clickthrough volume stays consistent at around 0.8 clickthroughs per query before filtering, and around 1.2 clickthroughs per query after filtering.

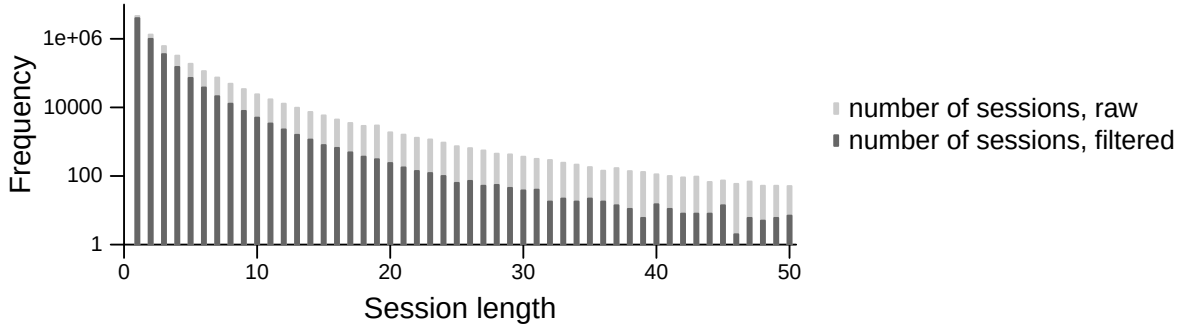


Figure 3: Frequency distribution for sessions of length 1 to 50, before and after the filtering process. Short sessions are more common than long sessions, a similar pattern witnessed in many other datasets. The filtering process removes queries, and thus tends to shorten sessions.

5 Sessions

Figure 3 portrays the distribution for sessions of different lengths, both before and after the filtering process. As is typical of other web search query sets, session are typically very short and contain just a few queries (less than two queries per session on average, for both raw and filtered query sets), and indicate relatively brief exchanges between users and the search engine. In the case of the filtered dataset, parts of the distribution gets shifted left as each session is trimmed of queries without any clickthroughs, resulting in much smaller numbers of longer sessions.

Figure 4 then shows the time difference in seconds between consecutive pairs of queries in multi-query sessions, using only the filtered query log. The majority of queries are issued within around one minute of each other, with very few queries issued at small time intervals (which, when it occurs, is a telltale sign of a session being machine driven). The largest interval recorded is $86{,}397$ seconds or just 3 seconds under 24 hours, which is perhaps indicative of the upper bound for the sessionization heuristic when the log data was prepared for distribution by Microsoft. Interactions over such a long period should probably not be considered as a single session.

The calculation used to determine resemblance is based on work by Broder [1997]. We define *resemblance* $R(A, B)$ between two queries A and B as:

$$R(A,B) = \frac{|S(A,n) \cap S(B,n)|}{|S(A,n) \cup S(B,n)|}$$

where $S(D, n)$ is the multiset of substrings of length n in the string D, not permitting any whitespace characters. In our calculations, we used $n = 3$ to obtain character *trigrams*. For example,

$$S(\text{“}\texttt{eat at the theater}\text{”}) = \{\text{“}\texttt{eat}\text{”}, \text{“}\texttt{the}\text{”}, \text{“}\texttt{the}\text{”}, \text{“}\texttt{hea}\text{”}, \text{“}\texttt{eat}\text{”}, \text{“}\texttt{atr}\text{”}, \text{“}\texttt{ter}\text{”}\}.$$

Figure 5 shows the distribution of multiset resemblance scores between consecutive queries in multi-query sessions for the filtered query log. Most follow-on queries bear relatively little resemblance to their predecessor, except in the special case when resemblance is 1.0. A resemblance of 1.0 means that it is highly likely that an identical query was submitted consecutively; and this happens when the user requests the “next” results page.

6 Queries

One of the great fascinations with query logs is to see what it is that people are searching for. A startling dis-

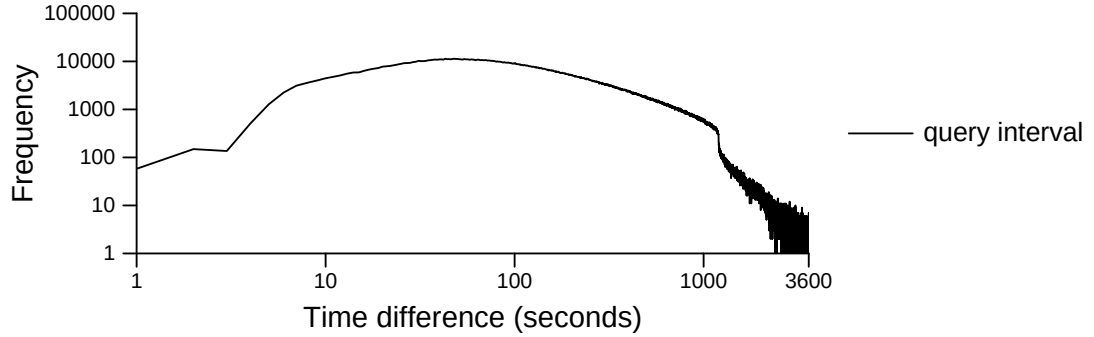


Figure 4: Interval in seconds between queries issued within a session, where sessions are as defined by Microsoft. The majority of intervals between same-session queries is less than a minute (60 seconds), although intervals of up to twenty minutes (1,200 seconds) are not uncommon. Only the filtered query log is shown in this graph. Note that the time intervals are quantized at one second values, but plotted as if they were continuous data.

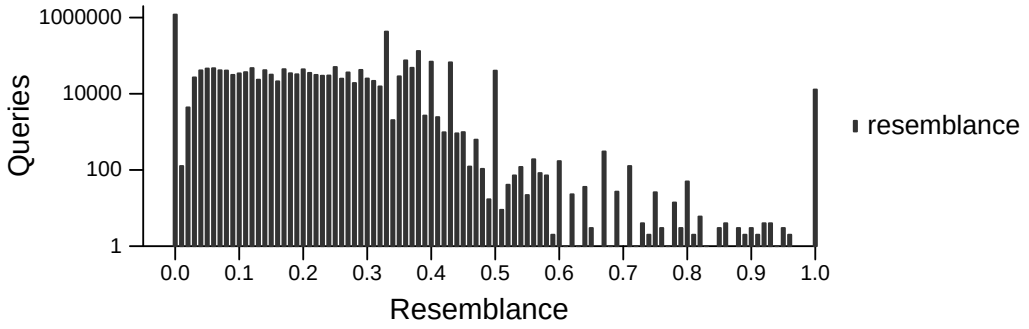


Figure 5: Trigram resemblance between consecutive queries issued within a session based on multiset overlap of trigrams of the query strings after filtering. A resemblance of 1.0 indicates that the pair of queries are identical; a resemblance of 0.0 occurs when the two queries have no trigrams in common.

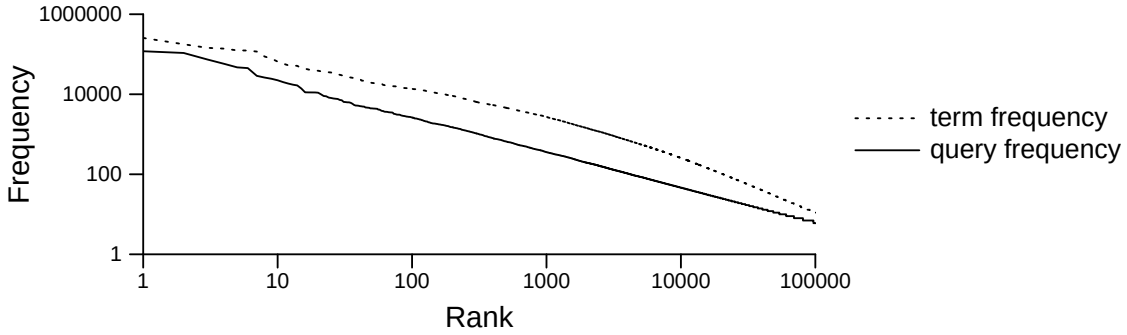


Figure 6: Rank-frequency distribution for queries and terms, in both cases after filtering. Both the query and term distributions follow the usual distribution. Note that neither queries nor terms were altered in any way when generating this figure, and we did not apply any stopping or case-folding techniques when creating the frequency distributions shown in the graph.

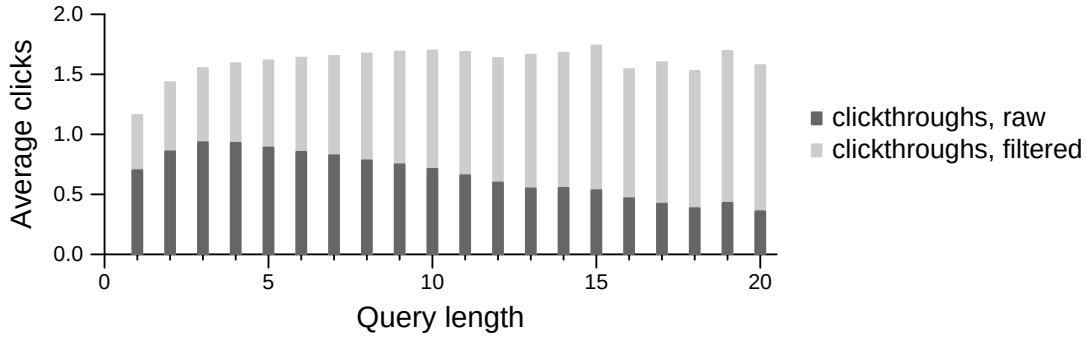


Figure 7: Query and clickthrough rates as a function of query length, measured as the average number of clickthroughs per query. Queries of more than ten terms have a reduced fraction of clickthroughs, indicating a possible lower availability of resultant data, or that (as is assumed in the filtering step) that these queries were automatically generated.

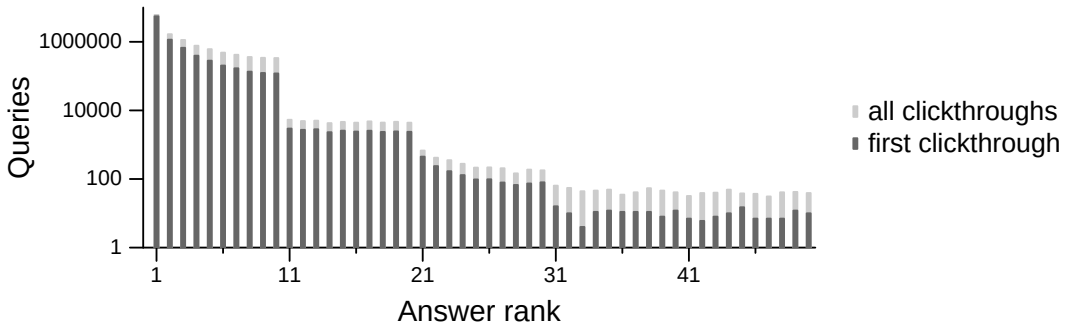


Figure 8: Positions in the results page at which clickthroughs occur, counting both the position of the first clickthrough per query, and the aggregate over all clickthroughs per query.

covery we made in this project is that when a query dialog box is available in a visible position (as it is, for example, in the MSN home page used by people to access hotmail accounts), its most common use is for navigational queries. The fifteen most popular queries in the query log supplied by Microsoft were all requests for other popular web services (including other search services), many specifying an almost full URL. For example, the query "`yahoo.com`" occurred more than $57{,}000$ times in the filtered query set, and was the 4th most common query. These top fifteen queries added up to 7.2% of the filtered query log, and it would appear from this snapshot – rather dishearteningly for academic IR researchers – that canned answers are probably the best way to respond to these queries. Also somewhat disheartening is that the sixteenth most popular query, and the first non-web-service one, was "`american idol`". (Note that the Microsoft asset includes a separate log of "adult" queries, and that the log we have used in this paper is the sanitized one. Determining the extent to which the Microsoft cleaning process alters query and response characteristics is left to others.)

Figure 6 shows the distribution of both whole queries, and terms within queries, taking frequency as a function of rank in the usual manner. The most frequent individual query term was the word "`of`", with "`in`" the second most common term. In both the raw and filtered query logs the most frequent non-web-service term and non-contentless term was the word "`county`".

7 Clickthroughs

One of the reasons why the Microsoft data is of great interest is because of the supplied clickthrough logs. Figure 7 shows the average number of clickthroughs per query for the raw and the filtered data. In the raw data, a clickthrough after a long query is relatively unlikely, decreasing as the query gets longer. In the filtered query set, the clickthrough rate is relatively constant across the range of query lengths for queries longer than five terms. For short queries – which represent the majority – the clickthrough rate is lowest on queries of length one. Given the nature of many of the short queries, discussed in the previous section, this is plausible – the query "`mapquest`" is highly likely to generate exactly one clickthrough, for example.

Figure 8 shows the position at which clickthroughs occur. The answer in rank position one is the most likely to be clicked. There is then a gradual drop in likelihood of a clickthrough through the rest of that first page, followed by a marked drop in the probability of any document beyond rank 10 being clicked. This pattern confirms that users are relatively unwilling to examine a second or subsequent results page via a "next"

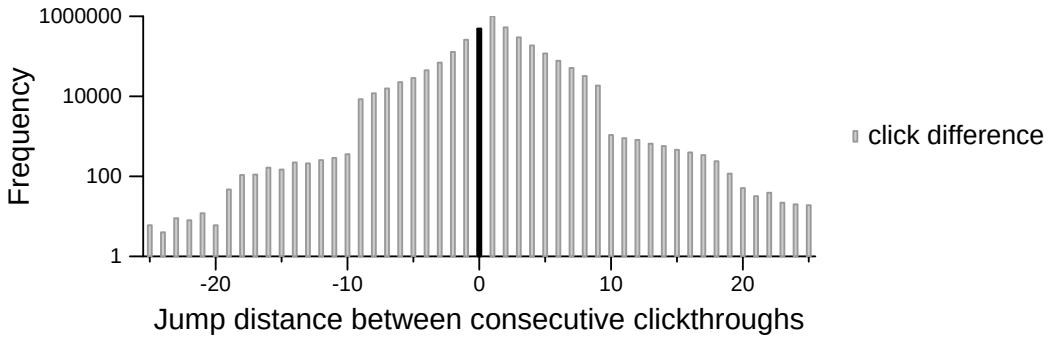


Figure 9: Jumps in clicked answer rank for queries that have two or more clickthroughs. The most common clickthrough jump is $+1$, to step from one proposed answer to the next.

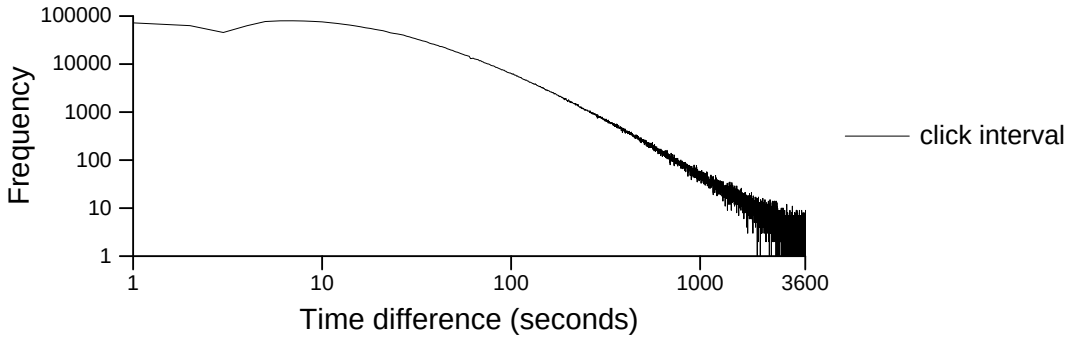


Figure 10: Interval in seconds between clickthroughs from a user for multi-clickthrough queries. Note that the time intervals are quantized at one second values, but plotted as if they were continuous data.

button, and that, within the first results page, preference is given to the answers that are presented near the top. Somewhat surprising in this graph is that there is a non-trivial minority of queries/users for which the first clickthrough does not take place until the third or even fourth results page for the query has been requested.

Figure 9 sheds further light on the manner in which users pursue paths through the presented data, by analyzing the sequence of clickthroughs on queries with more than one clickthrough. The most common clickthrough jump is $+1$, to step from one proposed answer to the next, as might be expected. But users are also nearly equally willing to backtrack through the results page, and click earlier answers, as they are to move forwards through the results pages. Users also (somewhat surprisingly) often click on the same answer document as consecutive actions, a jump of zero, perhaps caused by impatience, as they wait for a slow page to load. The sharp drops in frequency at jumps of -10 and $+10$ reinforces the fact that users are reluctant to examine subsequent result pages.

Our final graph, Figure 10, plots the time interval between consecutive clickthroughs for queries that generate multiple clickthroughs. Many user decisions to back out of one page, and clickthrough to another, are made within just a few seconds, and the decision time is typically less than a minute. This represents a quite different temporal distribution to the time intervals between queries in a session (Figure 4). Even unsophisticated users appear to have the ability to rapidly assess a page's relevance to them.

8 Related work

A recent review by Jansen and Spink [2006] provides a comprehensive overview into research activities in various fields of computer science utilizing different query logs, and compares key statistics over several significant studies prior to 2002. Several key outcomes regarding user browsing activity as well as syntax preferences were presented, although the authors noted the difficulty in drawing comparisons between studies involving different datasets.

One of the earliest log studies was conducted by Silverstein et al. [1999], who explored a query log containing approximately one billion queries from the AltaVista search engine and collected over a 43 day period, which is generally regarded as the largest dataset of its kind to date. The authors reported key statistics regarding query and session distributions, as well as significant query-term correlations. Silverstein et al. also noted that their dataset was not filtered to remove queries from automated sources.

In a similar experiment, Spink et al. [2001] examined a query log from Excite, comprised of over one million queries. The study found that typical queries are quite short and users generally only look at a few answer pages. Additionally, the authors provided a snapshot of query distribution in terms of topics, and discov-

ered that content within queries does not reflect the content available on the web. Lempel and Moran [2003] utilized another AltaVista query log containing around 7.7 million queries as part of their research into improving search engine throughput by caching popular query results. In this case, the statistics reported were focused towards patterns of page views.

There have been few large-scale studies into large volumes of clickthrough data, and it is in this respect that we feel our current work provides a contribution. Our results here can be seen as supporting recent work in connection with implicit relevance feedback, which contain some limited statistics regarding clickthrough outcomes [Agichtein et al., 2006, Joachims et al., 2005]. In this paper we have combined analysis of queries and clickthroughs in tandem, in order to draw out correlations between these two data streams.

9 Discussion and future directions

Much of what we have presented here simply confirms what has been found on other query streams – that queries are short; that a few queries (often completely inane) are very frequent in the query stream; and that there is a lot of mechanized access to search services. However, we can also draw a number of additional observations based on the clickthrough logs:

- Long queries have a smaller clickthrough rate;
- Users will sometimes take long jumps between consecutive clicks, and are also almost as likely to move backward as forward;
- Users dislike going beyond the first results page;
- Users are capable of making quick decisions about pages they have clicked on; and
- Users may click on the same answer page immediately after they have just viewed it.

One key issue that we may not yet have properly dealt with is that of spam removal within the queries, and possibly also within the clickthroughs (something which we have not considered). Similarly, the issue of session segmentation also needs to be addressed in order to create sessions of a finer granularity with more information value. In the absence of definitive information about the intentions of the user, such distinctions will remain elusive.

The natural application of our evaluations is to apply the understanding gleaned to try and improve search quality. Research by Joachims [2002] and Joachims et al. [2005] has shown that in a controlled setting, clickthrough data can be used to form pairwise relevance judgments, which in turn can be used to extract feature vectors for determining relevance of unseen documents. We will seek to explore these and related themes, possibly including a user study so that we have knowledge of user intention.

Acknowledgments Microsoft Research provided the logs described in this paper, and associated funding, via their "Accelerating Search" Project. Andrew Turpin (RMIT University) provided helpful input.

References

E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th ACM SIGIR Conference*, pages 3–10, New York, NY, USA, 2006. ACM Press.

A. Broder. On the resemblance and containment of documents. In *Sequences'97: Proceedings of the Symposium on Compression and Complexity of Sequences*, pages 21–29, Los Alamitos, CA, USA, 1997. IEEE Computer Society.

M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. *ACM Trans. Inter. Tech.*, 3(1):1–27, 2003.

T. Fagni, R. Perego, F. Silvestri, and S. Orlando. Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data. *ACM Trans. Inf. Syst.*, 24(1):51–78, 2006.

B. J. Jansen and A. Spink. How are we searching the world wide web?: A comparison of nine search engine transaction logs. *Inf. Process. Manage.*, 42(1):248–263, 2006.

T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the Eighth ACM SIGKDD Conference*, pages 133–142, New York, NY, USA, 2002. ACM Press.

T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th ACM SIGIR Conference*, pages 154–161, New York, NY, USA, 2005. ACM Press.

R. Lempel and S. Moran. Predictive caching and prefetching of query results in search engines. In *WWW '03: Proceedings of the 12th International Conference on the World Wide Web*, pages 19–28, New York, NY, USA, 2003. ACM Press.

C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

A. Spink, S. Ozmutlu, H. C. Ozmutlu, and B. J. Jansen. U.S. versus European web searching trends. *SIGIR Forum*, 36 (2):32–38, 2002.

A. Spink, D. Wolfram, B. J. Jansen, and T. Saracevic. Searching the web: the public and their queries. *J. Am. Soc. Inf. Sci. Technol.*, 52(3):226–234, 2001.

R. W. White, I. Ruthven, and J. M. Jose. A study of factors affecting the utility of implicit relevance feedback. In *SIGIR '05: Proceedings of the 28th ACM SIGIR Conference*, pages 35–42, New York, NY, USA, 2005. ACM Press.

InexBib - Retrieving XML elements based on external evidence

Alexander H. Krumpholz
David Hawking
Information Retrieval Group
CSIRO ICT Centre
Canberra
Alexander.Krumpholz@csiro.au

Abstract

Creating a scientific bibliography on a given topic is currently a task which requires a great deal of manual effort. We attempt to reduce this effort by developing a tool for automatically generating a bibliography from a collection of articles represented in XML. We evaluate the use of elements around the references as anchortexts to improve search results. We find that users of the tool prefer lists generated using anchortext over those generated from the bibliography entry only and that the preference is statistically significant. We tentatively find no significant preference for results generated using paragraph as opposed to sentence level anchortext, but note that this finding may result from lack of sophistication in resolving text including multiple references.

Keywords Information Retrieval, XML, Element Retrieval, Bibliography

1 Introduction

Over recent years XML has become a standard data exchange and storage format in all application domains. The 'INitiative for the Evaluation of XML Retrieval' (INEX) [5] studies XML retrieval techniques and evaluation methods. Its approach is focused on the retrieval of XML elements specified in the query (Content-and-structure (CAS) queries) or those best matching the search terms (Content-only (CO) queries), but in practice in this application, searchers usually prefer to retrieve whole documents. Indeed researchers have struggled to find motivating examples for element retrieval. We propose the construction of a reference list for a given topic as such a task, using the text around the reference in a publication analogously to anchortext in web retrieval to increase the retrieval quality.

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.

We present the results of a pilot study comparing the perceived quality of bibliographies generated with and without the use of 'anchortext'.

2 Related work

Previously published work in the areas of bibliometrics and the exploitation of anchortext in Web search are somewhat relevant to the present study as are systems such as CiteSeer[1] and Google Scholar[2].

2.1 Use of anchortext in retrieval

The usage of anchortext (the text actually forming the clickable link on web pages) has long been used to increase the retrieval quality of web search engines [12, 2, 3, 4]. Analogously to anchortext we indexed the text surrounding references as additional text for the bibliography entry. In two experiments we extracted the embedding sentence and paragraph respectively to explore the impact of context sizes on the retrieval quality..

Note that the use of anchortext introduces a voting effect. Bibliography items which are cited multiple times with descriptions matching the query will be ranked more highly than less frequently cited items.

2.2 Bibliometrics and bibliography generation

The field of bibliometrics [6, 14, 7, 10] concerns itself with the graph of citation links between scientific articles and has provided inspiration for link-based ranking methods in Web IR (e.g. [2]) but does not take account the descriptive text in citations.

CiteSeer [1] started to index publications and citations in 1998 and has become a widely used resource. Over time the CiteSeer database has grown to 730,000 documents with over 8 million citations and a new version CiteSeerX has recently been presented [11]. CiteSeer has access to a much larger

[1] `http://citeseer.ist.psu.edu/`
[2] `http://scholar.google.com/`

database of citations than we are using, and can be used to retrieve a list of references that match query keywords. However, we are not aware that CiteSeer uses descriptive anchortext in the retrieval process.

No other publication known to the authors investigates anchortext approaches for reference list generation.

3 Method

In this section we characterize the INEX data, and explain how we extracted bibliographic items and matched them to citations in the articles. We then describe the retrieval software we used and how we built the three different indexes used in the study.

3.1 INEX data

The data corpus used by INEX for the last four years is a collection of over 12,000 journal articles from 18 IEEE journals from years between 1995 and 2002. (See tables 1 and 2.)

The articles are stored in XML format as described by Fuhr et al. in [5], allowing researchers to develop and apply XML retrieval techniques to create the result lists defined for the current INEX round.

Listing 1 shows an example reference, Listing 2 an example bibliography entry.

In order to retrieve elements other than those whose bibliography entry matched the search terms, we extracted almost 150,000 references from the bibliographies of all the journal articles in the collection, saved them into separate files and used naive record linkage techniques to identify publications cited by multiple articles.

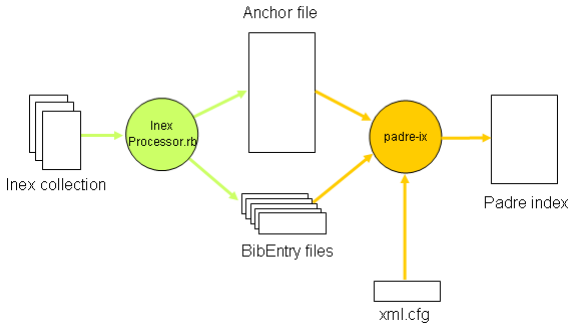


Figure 1: Preprocessing

```
...
A more detailed discussion on
fairness can be found in
[<ref rid="bibL03371" type="bib">1</ref>],
[<ref rid="bibL033710" type="bib">10</ref>].
</p>
...
```

Listing 1: Example reference

```
... <bb id="bibL03371">
  <au>
    <fnm>K.R.</fnm>
    <snm>Apt</snm
  </au>
  <au>
    <fnm>N.</fnm>
    <snm>Francez</snm>
  </au>
  <obi>and</obi>
  <au>
    <fnm>S.</fnm>
    <snm>Katz</snm>
  </au>
  <atl>&ldquo;Apprasing Fairness in
       Languages for Distributed
       Programming,&rdquo;</atl>
  <ti>Distributed Computing,</ti>
  <obi>
    <volno>vol. 2,</volno>
  </obi>
  <pp>pp. 226-241,</pp>
  <pdt>
    <yr>1988.</yr>
  </pdt>
</bb> ...
```

Listing 2: Example bibliography entry

3.2 Record linkage

The INEX collection contains the bibliography entries and the references already extracted into XML elements and linked via artificial keys. However, as the keys are only unique with one journal article, multiple articles refer to the same publication using different keys. Since bibliography entries are often referenced by multiple journal articles, a deduplication problem had to be addressed. In order to prepare for the bibliography entries to be indexed, each entry has been extracted into a separate file.

The document type definition of the journal articles has been defined very vague to allow all possible bibliography entries to be recorded; however, this allows for multiple or missing elements, which made the record linkage a non-trivial task.

We introduced a key for bibliography entries by combining the first author's last name with the title of the publication to identify references of identical publications. However, even though we converted all characters to lowercase, removed special characters and entities like '“' (see Listing 2), some

Period	Journal
1995 – 2001	IEEE Annals of the History of Computing
1995 – 2001	IEEE Computer Graphics and Applications
1995 – 2001	Computer
1995 – 2001	Computing in Science and Engineering
1995 – 2001	IEEE Design & Test of Computers
1995 – 2001	IEEE Intelligent Systems
1997 – 2001	IEEE Internet Computing
1999 – 2001	IT Professional
1995 – 2001	IEEE Micro
1995 – 2001	IEEE MultiMedia
1995 – 2000	IEEE Concurrency
1995 – 2001	IEEE Software
1995 – 2002	IEEE Transactions on Computers
1995 – 2002	IEEE Transactions on Parallel and Distributed Systems
1995 – 2002	IEEE Transactions on Visualization and Computer Graphics
1995 – 2002	IEEE Transactions on Knowledge and Data Engineering
1995 – 2002	IEEE Transactions on Pattern Analysis and Machine Intelligence
1995 – 2002	IEEE Transactions on Software Engineering

Table 1: Journals

errors have been found within the data that did not allow some records to be linked correctly.

One surname element for example contained *Hudak at al.*, another one the string *Agrawaland* for an author named *Agrawal*, obviously containing the *and* from the list of authors. Some publications did not have an author specified at all. Altogether 10,382 of the bibliography entries do not have an author specified and got a key using the string 'UNKNOWN' as the author's last name.

Titles have not always been cited correctly, e.g.

```
3 Weighted Pseudo Random Test Generation
3 Weight Pseudo Random Test Generation
```

and 5,136 entries (3.4%) have been skipped altogether, since no title is defined.

Stemming or even probabilistic record linkage techniques could be used to increase the number of correctly identified publications but this potentially introduces false positives and for the scope of this prototype we accepted that some links would be missed, even though this can cause duplicate entries in the reference lists generated by our prototype.

3.3 Retrieval engine

In our experiments, we used the PADRE retrieval system [8]. For text ranking PADRE uses a marginally modified Okapi BM25 relevance function developed by Robertson et al. [13]. PADRE makes use of anchortext extracted from web documents as they are indexed and is also capable of using externally derived anchortext files. Anchortext scoring uses the AF1 formula described by Hawking et al. in [9].

In a second processing iteration we extracted all elements containing one or more references within the text for each journal article.

In addition to using the whole parent element of a reference ($< ref >$) element – usually a paragraph ($< p >$) – as anchortext, we extracted the sentence containing the reference for an alternative index. Sentences are defined as full stop delimited areas around a reference. From these extracted elements we created anchortext records for each reference. In each record the link target was the key we assigned to the reference and the anchortext was either the surrounding sentence or the surrounding paragraph.

We decided to build three different indices using different sources of anchortext:

- AtN - Anchortext not used
- AtS - Anchortext based on Sentences
- AtP - Anchortext based on Paragraphs

The first used no anchortext and the others used sentence-level and paragraph-level anchortext respectively.

To increase precision at search time, and to reduce the length of reference lists to be judged, we configured our search engine to only display full matches. In some cases this caused generated lists to contain only a few or even no records.

Table 2 gives some details of the data indexed.

4 Experiments

In our evaluation the quality of reference lists generated from the three indices was evaluated by fif-

	Quantity	Size (MB)	Size of tar (MB)
Articles	12,107	536	
BibEntries	149,168		
BibEntries with unknown author	10,382		
BibEntries used article title	121,971		
BibEntries used publication title	22,061		
BibEntries skipped (no title)	5,136		
BibEntries files created	96,491	476	140
BibEntries files reused	47,541		
References	241,228		
References without a refid	11		
References found its BibEntry	233,602		
References found no BibEntry	7,615		
Anchortext file paragraph		205	
Anchortext file sentence		85	

Table 2: Preprocessing quantity structure

teen experimental subjects (all researchers from our institution who volunteered to participate) using a comparative approach. The tool used to do the comparison was based on one described by Thomas and Hawing in [15].

After consenting to participate and logging in the subjects were given the task to generate bibliographies by entering topics and to judge the reference lists returned. The subjects were presented with an interface including a search box and were encouraged to enter a query representing a research topic in which they were interested. In response, the comparison tool presented three results lists generated by processing the query against each of the three indices. As can be seen in Figure 2, results were presented in normal bibliography style. The three lists were displayed side-by-side in random order for each query to avoid a bias for or against particular screen locations. Users were able to choose the length of results list before searching and also to request a longer or shorter list at any time during the process. For each of the three bibliographies, the users were asked to assess quality on a scale of 0 to 9 (0/useless - 9/excellent). With each query they were also invited to store a comment on the judging.

All judgments and comments were recorded for analysis.

5 Results

To compare the three versions, we used AtN as our baseline system and compared the subject's judgements relative to that baseline.

The quality of the reference lists generated subjectively varies for different queries and the subjects were allowed to pick their own topics.

User	Judgements		
	AtN	AtS	AtP
User A	1	5	6
User B	0	8	7
User C	2	8	7

Table 3: Different judgements for query 'haptics'

Table 3 shows that subjects judged the results for coincidentally identical queries differently, not only in absolute figures but also relative to each other.

A statistical analysis of variance (ANOVA) of the collected data shows that the participating research scientists preferred the anchortext versions to the plain bibliography entry index with statistical significance with a p-value $\leq .001$ using Fisher's Least Significant Difference (LSD) test (using 242 degrees of freedom). However, no significant preference between the sentence and the paragraph based approach could be identified.

Figure 4 shows the total number of cases, in which the subjects preferred AtP to AtN. Subjects often ranked AtP equally to the baseline, but when they perceived a difference it was most often in favour of AtP. The same comment is valid for Figure 5. AtS is preferred to the baseline.

Figure 6 shows the comparison between AtP and AtS. The values are normal distributed around zero, which visualized the results shown above; the difference between AtS and AtP is not significant.

Figure 3 shows for each subject the mean preference for the sentence or paragraph based version versus the baseline that uses no anchortext at all. In all but one case the anchortext approaches were preferred. Only subject 9 found the list generated

using a paragraph based anchortext approach on average worse (by one point) than the one generated by the baseline system.

6 Discussion

The three different lists presented show the typical tradeoff between precision and recall. The baseline version always exactly matched against the bibliography entry, while the sentence based anchortext approach increased recall at the cost of precision. Some searchers observed, that some entries returned by AtS and AtP do not contain the search term.

To increase the precision we configured our search engine to only display full matches. This caused some of the generated lists to only contain a few or even no records.

The paragraph extracted is defined as the super element of the text containing the reference. Sometimes this super element was not a paragraph element, but for example a table data element. Sentences are defined as full stop delimited areas around a reference.

In the current version, the complete sentences and paragraphs containing multiple references are indexed for each of the references. This might be the source of wrong mappings in cases where a single sentence or paragraph refers to different topics. Developing more sophisticated algorithms to split paragraphs and sentences into units of text more relevant to the referenced publication is expected to increase quality.

Depending on the purpose of the reference list created, different precision/recall tradeoffs might be preferred.

Reference lists generated by our system are presented in order of descending scores assigned by the retrieval system. Anchortext ranking tends to mean that highly cited items will be highly ranked, which is probably an advantage. However, researchers may prefer a date or author ordered listing.

Our subjects didn't always look at the entire reference lists when making their ratings, suggesting that they made their judgments based on early precision or on the presence or absence of expected key items at the head of the list. This implies that the ranking of references is important and that in future work, it would be worth paying attention to optimizing the ranking function for this specialised purpose.

6.1 Alternative evaluation approaches

Our subjects were able to judge the quality of the returned lists on a purely subjective basis. This has its advantages but in future work, we will consider both asking subjects to judge the complete reference lists and asking them to additionally rate the value of each bibliography item. We envisage modifying the three-panel comparison tool to allow such judgments and to set a background colour for each judged item wherever it appears in each list.

As an alternative evaluation approach, a comparison between a survey paper[3]'s bibliography and a reference list automatically generated using our methods was considered. However, this evaluation technique was not selected for fear of restrictively limited overlap between the selected survey paper's bibliographies and papers cited by the IEEE articles in the matching time frame. This method may be able to provide a more absolute type of judgment and will also be considered in future work.

6.2 Biases

The subjects in the pilot experiment were mainly research scientists from within our institution. A different set of subjects might judge the quality of the lists differently. However, it should be noted that the target group of a tool generating reference lists is not the general public.

The nature of the approach and the age of the data set mean that work published after the most recent articles in the INEX collection cannot possibly be retrieved by our system. The only solution to this seems to be to endeavour to obtain more recent data. The tendency for citation counts to increase with the passage of time since publication also means that our anchortext rankings are likely to rank older items higher in the list.

7 Conclusions

Using a three-way side-by-side comparison of reference lists automatically generated from the INEX collection, we have shown that anchortext techniques from web retrieval are also beneficial in the XML retrieval domain. As yet, it is not clear whether the anchortext scope should be at the sentence or paragraph level. Ideally, natural language processing techniques might be used to set the appropriate context for each reference, particularly where multiple references occur within the same paragraph or sentence.

Our tool for generating reference lists from a collection of scientific articles illustrates a useful application for the retrieval of elements other than full documents, and an application that retrieves XML elements based on data outside of that element or its sub tree.

Many avenues have been identified for possible future work, including investigation of alternative evaluation methods, better anchortext extraction and the use of more extensive and up-to-date data. It would be very interesting to study when and

[3]e.g. those in ACM Computing Surveys.

how searchers would actually use a bibliography generation tool if they had access to one, and to perform more thorough evaluation of the tool in the context of real bibliography generation tasks.

Acknowledgements

We want to thank Anne-Marie Vercoustre of INRIA for various discussions and feedback and Alec Zwart of CSIRO Mathematical and Information Sciences for statistical analyses. Finally we'd like to thank the participants in this experiment for their time and efforts.

References

[1] Kurt Bollacker, Steve Lawrence and C. Lee Giles. CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications. In Katia P. Sycara and Michael Wooldridge (editors), *Proceedings of the Second International Conference on Autonomous Agents*, pages 116–123, New York, 1998. ACM Press.

[2] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of WWW7*, pages 107–117, 1998. `www7.scu.edu.au/programme/fullpapers/1921/com1921.htm`.

[3] Nick Craswell, David Hawking and Stephen Robertson. Effective site finding using link anchor information. In *Proceedings of ACM SIGIR 2001*, pages 250–257, New Orleans, 2001. `www.ted.cmis.csiro.au/nickc/pubs/sigir01.pdf`.

[4] B. Davison. Topical locality in the web. In *Proceedings of ACM SIGIR'2000*, pages 272–279, Athens, Greece, 2000. `www.cs.rutgers.edu/~davison/pubs/2000/sigir/`.

[5] Norbert Fuhr, Norbert Gövert, Gabriella Kazai and Mounia Lalmas. INEX: INitiative for the Evaluation of XML Retrieval. In *Proceedings of the SIGIR 2002 Workshop on XML and IR*, 2002.

[6] E. Garfield. Citation analysis as a tool in journal evaluation. *Science*, Volume 178, pages 471–479, 1972.

[7] N. Gilbert. A simulation of the structure of academic science. *Sociological Research Online*, Volume 2, Number 2, 1997.

[8] David Hawking, Peter Bailey and Nick Craswell. Efficient and flexible search using text and metadata. Technical Report TR2000-83, CSIRO Mathematical and Information Sciences, 2000. `http://www.ted.cmis.csiro.au/~dave/TR2000-83.ps.gz`.

[9] David Hawking, Trystan Upstill and Nick Craswell. Towards better weighting of anchors (poster). In *Proceedings of SIGIR'2004*, pages 99–150, Sheffield, England, July 2004. `http://es.csiro.au/pubs/hawking_sigirposter04.pdf`.

[10] R. Larson. Bibliometrics of the world wide web: An exploratory a nalysis of the intellectual structure of cyberspace. In *Annual Meeting of the American Society Information Sc ience*, 1996.

[11] Huajing Li, Isaac Councill, Wang-Chien Lee and C. Lee Giles. Citeseerx: an architecture and web service design for an academic document search engine. In *Proceedings of WWW 2006, May 2326, 2006, Edinburgh, Scotland.*, 2006.

[12] Oliver A. McBryan. GENVL and WWWW: Tools for Taming the Web. In *ProceeProceedings of the First International World Wide Web Conference 1994*, 1994.

[13] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu and M. Gatford. Okapi at TREC-3. In D. K. Harman (editor), *Proceedings of TREC-3*, Gaithersburg MD, November 1994. NIST special publication 500-225.

[14] H. Small. Co-citation in the scientific literature: A new measu re of the relationship between two documents. *Journal of the American Society for Information Scien ce*, Volume 24, 1973.

[15] Paul Thomas and David Hawking. Evaluation by comparing result sets in context. In *To appear in Proc. CIKM 2006*, 2006.

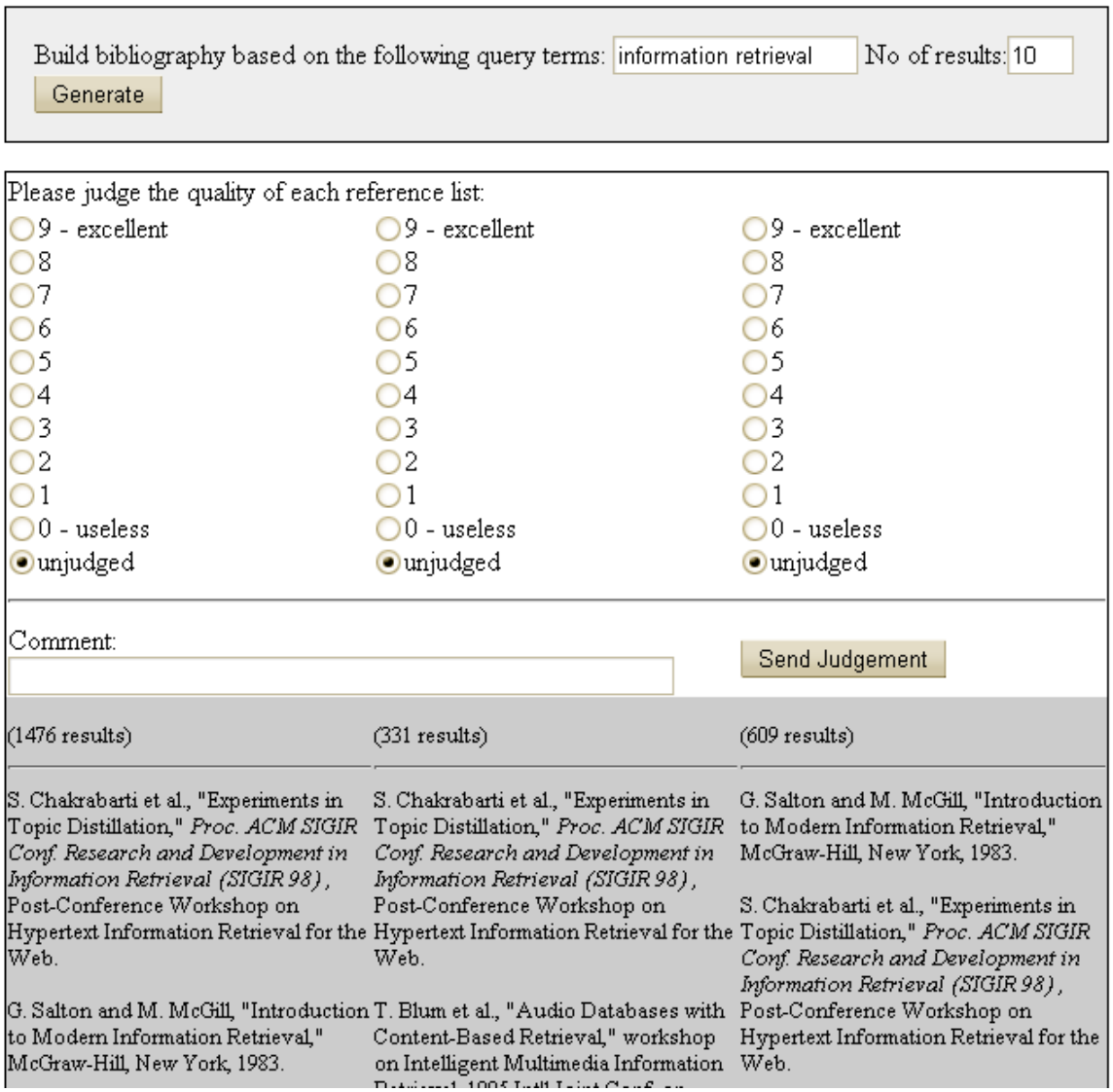


Figure 2: Judging interface

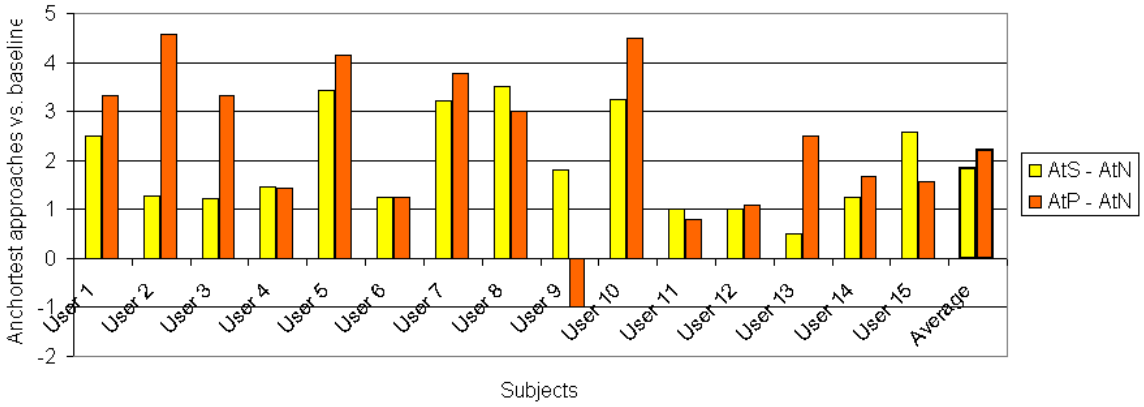


Figure 3: User Comparison

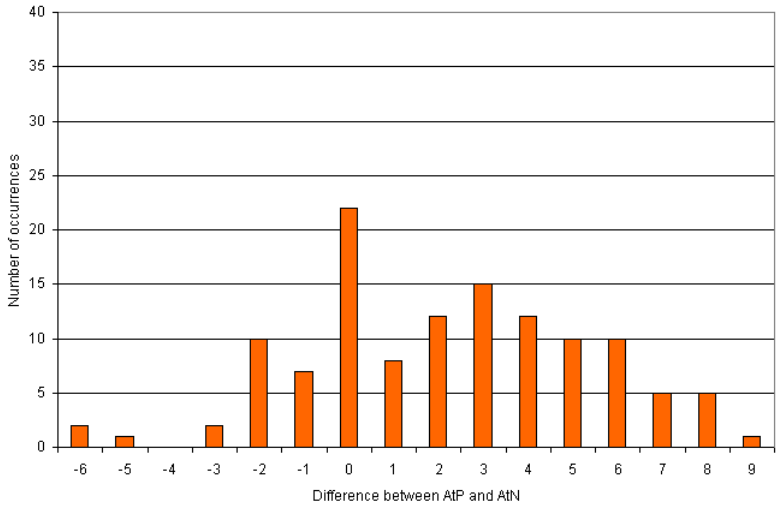


Figure 4: Distribution of AtP rating minus AtN rating

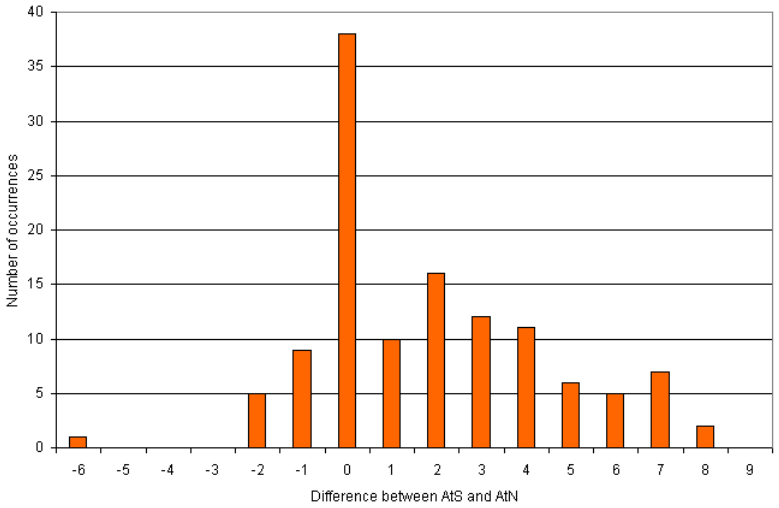


Figure 5: Distribution of AtS rating minus AtN rating

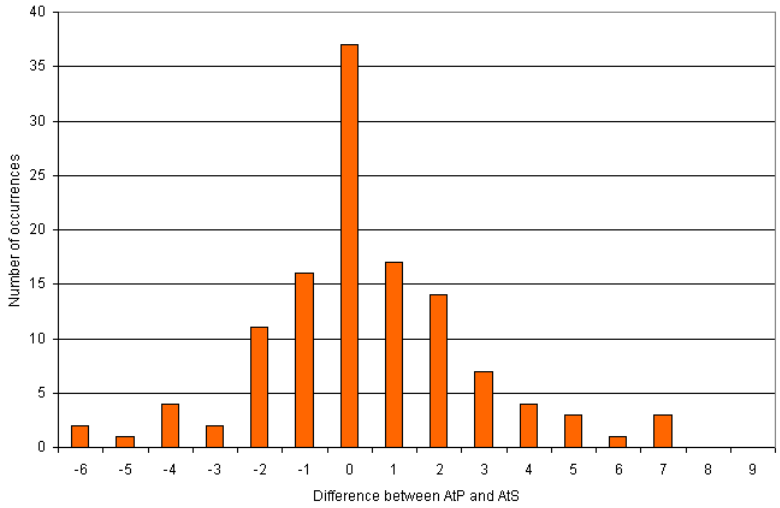


Figure 6: Distribution of AtP rating minus AtS rating

Element Retrieval Using a Passage Retrieval Approach

Weihua Huang
rory_huang@hotmail.com

Andrew Trotman
andrew@cs.otago.ac.nz

Richard O'Keefe
ok@cs.otago.ac.nz

Department of Computer Science
University of Otago
Dunedin, New Zealand

Abstract

Element and passage retrieval systems are able to extract and rank parts of documents and return them to the user rather than the whole document. Element retrieval is used to search XML documents and identify relevant XML elements, while passage retrieval is used to identify relevant passages. This paper reports a series of experiments on element retrieval, using a general passage retrieval algorithm. Firstly, an XML document is divided into overlapping or non-overlapping fixed size windows (passages), then the relevant passages which contain query terms are found. Given the position of a passage in the XML document, the smallest element which contains this passage is found. The experiments were conducted with the INEX 2005 ad hoc test collection and evaluation tool. Two passage extraction methods, three weight functions and various window sizes were tested. A comparison with element retrieval systems was also conducted. The experimental results show that a robust passage retrieval algorithm can yield an acceptable level of performance in XML element retrieval.

Keywords Element retrieval, passage retrieval, XML retrieval, INEX.

1 Introduction

Both element and passage retrieval are able to extract and rank small relevant parts of a long document, which addresses some shortcomings of traditional whole-document retrieval systems. Element retrieval is used to search XML documents and to identify relevant XML elements. Passage retrieval is used when there is no mark-up — these algorithms identify relevant passages of text.

Element retrieval relies on the structure of XML documents in which the content is organized into smaller, nested structural elements. Each of these elements in the document's hierarchy, along with the document itself (the root of the hierarchy), is a retrievable unit [4]. Due to the nested hierarchical structure of XML documents, a query of an XML retrieval system can be expressed as a combination of content and structural conditions.

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.

Passage retrieval is the task of identifying and extracting fragments from heterogeneous full-text documents. A retrieved passage can be one or more sections, paragraphs, sentences, or a fixed number of words.

Our element retrieval system employs a passage retrieval algorithm that divides an XML document into passages by sliding a fixed size window across the document. The main purpose of this paper is to investigate the behaviour of the passage retrieval technologies for element retrieval, and then compare this approach with other element retrieval systems. Specifically, we want to compare the performance of previous passage retrieval algorithms (that ignore document semantics) with element retrieval algorithms (that do not). We compare our implementation to those of consistent good performers at INEX.

2 System Overview

The experiments are conducted as follows. Each XML document is divided into fixed size overlapping or non-overlapping windows (passages). If a window contains at least one query term, then it is a relevant passage. Given the element paths of the starting word and finishing word of this relevant passage, their common element ancestor is the smallest XML element that fully contains the passage. This element is then given the retrieval status values (RSV) of the passage. Overlapping elements are removed and, for each query, the first 1500 most highly scored elements are output as an INEX submission file. The score of a run is computed using the INEX assessment and evaluation tools.

We envisage a two pass retrieval system. First relevant documents are identified, then from that pool, relevant fragments are identified. So, in order to compare passage and element retrieval *on the same documents*, only those documents already identified by a search engine were examined. These documents were those identified by the IBM submission to INEX 2005, to which we compare our results.

The experiments investigated three weighting functions [1]:

2.1 Term Frequency Model (FREQ)

In this approach, the weight of each window is calculated by summing the total occurrences of all terms, t_i of a query Q within the window, W. The window RSV can be computed as follows [1]:

$$P(anyQterm|W) = \sum_{t_i \in Q} p(t_i|W)$$

The FREQ weighting approach is used as a baseline.

2.2 Query Generation Model (GEN)

In this approach the window RSV is computed as the probability of generating a query[1]:

$$P(Q|W) = \prod_{t_i \in Q} p_{mix}(t_i|W)$$

where

$$p_{mix}(t_i|W) = \lambda \times p(t_i|W) + (1 - \lambda) \times p(t_i|D)$$

The mixing parameter, λ, is to smooth the estimates. In this work, λ is set to 0.8. In [1] Harper and Lee investigate the best value for λ and best results were obtained in the range 0.8 through 0.999. λ = 0.8 is the value they suggest using. The word probabilities are calculated as follows:

$$p(t_i|W) = n_{iW}/n_W \quad p(t_i|D) = n_{iD}/n_D$$

where $n_{iW}(n_{iD})$ and $n_W(n_D)$ are the number of term occurrences of term i in the window (document), and total term occurrences in the window (document) respectively. The log of both sides of the first formula is then taken:

$$\log P(Q|W) = \sum_{t_i \in Q} \log p_{mix}(t_i|W)$$

2.3 Kullback-Leibler Model (KL)

Using this approach, the window RSV is calculated as follows [1]:

$$KL(W|Q) = \sum_{t_i \in Q} p(t_i|W) \log(p(t_i|W)/p(t_i|D))$$

where

$$p(t_i|W) = (n_{iW} + 0.5)/(n_W + 1.0)$$

$$p(t_i|D) = (n_{iD} + 0.5)/(n_D + 1.0)$$

3 Test Set and Task

The test collection was the INEX 2005 *ad hoc* test set, which contains a document collection, a set of queries, and relevance assessments.

The INEX 2005 document collection comprises 16,819 scientific articles published between 1995 and 2004. It contains more than 10 million XML elements and is about 764MB in size.

With respect to the element retrieval task, the CO.Focussed sub-task was tested, which is to find non-overlapping relevant elements [2].

For the query set, the INEX 2005 CO query set containing 40 topics was used. All the queries used in the project contain exactly the same query terms as the <title> part of the topic. There are five queries that contain two words and others contain three or more words. There is no single word query.

This system did not apply stemming or use stop words which is likely to affect the results substantially.

To evaluate the system, the INEX 2005 official evaluation tool XCGEval was used. The official system-oriented evaluation was based on the *ep/gr* measures, with mean average (*MAep*) and interpolated mean average (*iMAep*) being the overall performance indicators. The experimental results were generated using generalised quantisation [2].

4 Experiment Objectives

Four questions are investigated:

- Could current passage retrieval algorithms be used to retrieve XML elements? If so then how well?

- A document can be divided into overlapping or non-overlapping passages. Which method produces better results?

- How does window size affect the performance of a passage retrieval system? Is there an optimal window size for the overlapping and non-overlapping window?

- The retrieval status value (RSV) of each window is computed by using a weighting function. Three passage weighting functions are investigated. Which one is best?

5 Experiment Results

5.1 Overlap Experiments

Two different passage extraction methods were investigated: overlapping windows and non-overlapping windows. The non-overlapping method is to divide a document into a set of fixed size non-overlapping blocks, such as pages. The overlapping approach, however, divides the document up into fixed size possibly overlapping blocks. For example, a heavily overlapping window may start from the second word of the previous window. Intuitively, this approach may report too many redundant results. To avoid this problem, the window is slid until the first word of the window is in the query. The overlapping window approach used ensures a reported window always begin with a query word, but need not end with a query word.

The overlapping and non-overlapping window results are presented in Figures 1, 2 and 3 for the best window sizes we discovered. Figure 1 shows the results of window size 100 on the FREQ weighting function. The overlapping window approach performs better than the non-overlapping one (MAep 0.0158 vs 0.0119), especially at the low recall levels. Figure 2 shows window

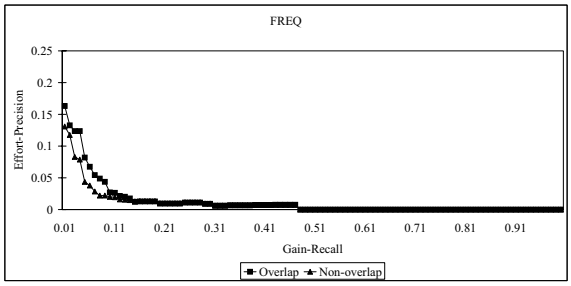


Figure 1: FREQ model, window size = 100

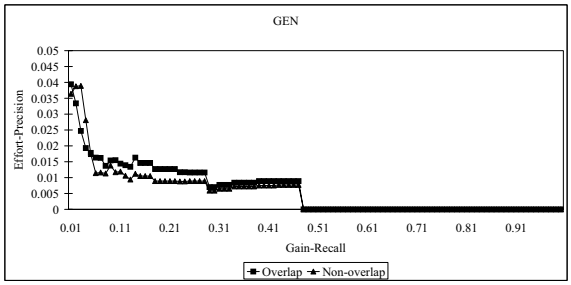


Figure 2: GEN model, window size = 250

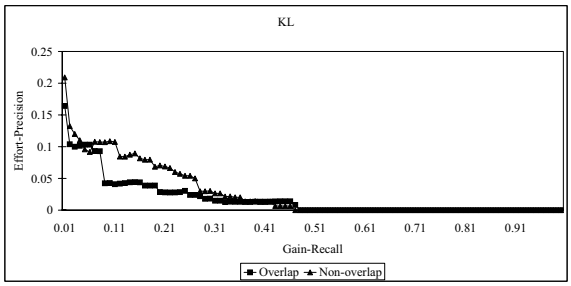


Figure 3: KL model, window size = 225

size 250 on the GEN model, in which the overlapping window approach also performs better than the non-overlapping one (0.0077 vs 0.0066).

Figure 3 shows window size 225 on the KL model. Surprisingly, on this model the overlapping window approach performs much worse than the non-overlapping one (0.0197 vs 0.0316).

5.2 Window Size Experiments

The second experiment investigated the effect of the window size on three weighting functions. The size of of the sliding window from 75 words up to 250 words increasing by 25 words each time.

This window size changing method was tested on three weight functions: the query generation model (GEN), the Kullback-Leibler model (KL), and the term frequency model (FREQ). The passage extracting method was overlapping windows.

The window size experimental results are presented in Tables 1, 2 and 3, where the **bold figures** are the best values achieved for each weighting function.

For the GEN model, smaller window sizes such as 75, 100 yield slightly better results than the bigger ones (125 – 200).

Table 2 and 3 show the performances of KL and FREQ model using overlapping passages. KL produces better results when the window size is large. On the other hand, FREQ model yields better results with small sized windows (75, 100).

A special window size test for KL model using non-overlapping passages was conducted. The results are shown in table 4.

When the run of 250 words size window finished, the experimental results showed that the KL model tends to favour large window sizes. To validate this conjecture, another six runs on window sizes from 275 to 400 were carried out. The results shows that KL produces better results when the window size is large.

window size	MAep	iMAep
75	**0.0077**	**0.0062**
100	**0.0077**	0.0061
125	0.0075	0.0058
150	0.0075	0.0057
175	0.0076	0.0059
200	0.0075	0.0060
225	**0.0077**	0.0061
250	**0.0077**	0.0060

Table 1: GEN_Model using overlapping passages

window size	MAep	iMAep
75	0.0149	0.0138
100	0.0169	0.0159
125	0.0178	0.0167
150	0.0183	0.0171
175	0.0186	0.0173
200	0.0194	0.0181
225	**0.0197**	**0.0182**
250	0.0182	0.0163

Table 2: KL_Model using overlapping passages

window size	MAep	iMAep
75	0.0145	0.0117
100	**0.0158**	**0.0124**
125	0.0138	0.0103
150	0.0138	0.0098
175	0.0138	0.0103
200	0.0130	0.0098
225	0.0131	0.0095
250	0.0129	0.0093

Table 3: FREQ_Model using overlapping passages

5.3 Weighting Function Experiments

In this section three weighting functions are compared. For comparing the performances of different weighting

window size	MAep	iMAep
75	0.0151	0.0137
100	0.0221	0.0204
125	0.0279	0.0259
150	0.0296	0.0267
175	0.0322	0.0294
200	0.0339	0.0303
225	0.0316	0.0277
250	0.0383	0.0337
275	0.0396	0.0344
300	0.0403	0.0349
325	0.0419	0.0361
350	0.0439	0.0365
375	0.0446	0.0371
400	**0.0454**	**0.0376**

Table 4: KL_Model using non-overlapping passages

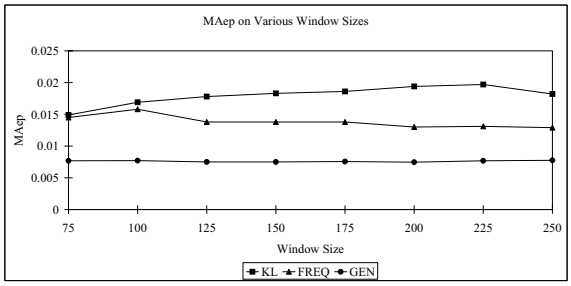


Figure 4: MAep on various window sizes

functions, the window size was fixed at various settings for the three weighting functions. The results are shown in figure 4.

Figure 4 shows that, for all the tested window sizes, the performance of the Kullback-Leibler model exceeds that of the other two weighting functions. Interestingly, it seems that FREQ performs better than GEN.

5.4 Comparison with Element Retrieval

The last experiment was to compare this system with element retrieval systems. Because the new system employed passage retrieval algorithms and passage weighting functions to retrieve XML elements, which has not been researched before (as far as we are aware), it is interesting to know how well the system performs by comparison to element retrieval.

To conduct this comparison, two sets of INEX 2005 submission files were chosen: IBM Haifa research lab (from which we took the relevant document list) which was ranked 4^{th} at INEX 2005 and University of Amsterdam which ranked 28^{th} (of 44).

The submission file for our system used non-overlapping passages. The window size was 300 words. The weight function was KL. We are aware that our results are overfitted and include them only to show that such an approach could be effective. No significance tests were conducted for the same reason.

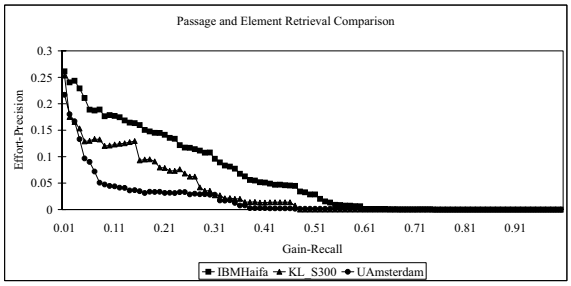


Figure 5: Comparison with element retrieval

Figure 5 shows that the run of IBM Haifa exceeds that of the other two systems. Our system performs better than that of the University of Amsterdam except at low recall levels.

6 Conclusion

This paper describes a comparison between passage and element retrieval approaches. The experiments were conducted with the INEX 2005 test collection and official evaluation tool. Two passage extracting approaches, three window weighting functions and various window sizes were tested. Major findings are:

- Given a robust passage retrieval algorithm and an XML parser, it is possible to retrieve elements efficiently.

- Compared with its non-overlapping counterpart, the overlapping window approach yields better results on the query generation model (GEN) and term frequency model (FREQ), but not on the Kullback-Leibler model (KL).

- There is a notable difference of performance when using different window sizes. The Kullback-Leibler (KL) favours larger windows.

- Among the three weight functions, the KL model outperforms the others. The query generation model (GEN) produced worse results than expected.

- Prior passage retrieval systems tuned for the document collection perform well compared to element systems tailored to the collection. It is reasonable to investigate tailoring a passage system to the collection and perhaps to include structural semantics in the algorithm.

References

[1] D. J. Harper and D. Lee (2004), On the Effectiveness of Relevance Profiling, *In Proceedings of the 9th ADCS*, pp. 10-16.

[2] G. Kazai and M. Lalmas (2005), INEX 2005 Evaluation Metrics, *In INEX 2005 Pre-proceedings*, pp. 401-406.

[3] Y. Mass and M. Mandelbrod (2004), Component Ranking and Automatic Query Refinement for XML Retrieval, *In Proceedings of INEX 2004*, pp. 73-84.

[4] B. Sigurbjörnsson, A. Trotman, S. Geva, M. Lalmas, B. Larsen and S. Malik (2005), INEX 2005 Guidelines for Topic Development, *In INEX 2005 Pre-proceedings*, pp. 375-384.

Differentiating Document Type and Author Personality from Linguistic Features

Scott Nowson
Centre for Language Technology
Macquarie University
NSW 2109 Australia
snowson@ics.mq.edu.au

Jon Oberlander
Division of Informatics
University of Edinburgh
Edinburgh EH8 9LW UK
j.oberlander@ed.ac.uk

Abstract

There are many ways to profile a collection of documents. This paper presents highlight from a body of work that has looked at individual differences in the language of personal weblogs. Firstly, we present a unitary measure of linguistic contextuality based on POS frequency that can be used to profile and rank genres. When applied to weblogs, we will show they are similar to school essays, yet significantly less contextual than e-mail. We then look at individual variation of language, as due to the personality of the author, exploring the use of dictionary based analyses and data-driven n-grams. Under regression, we show that with just a few linguistic features, it is possible to explain significant proportions of variance within personality traits.

Keywords Personalised Documents; Multimedia Resource Discovery

1 Introduction

With the increasing amounts of data available to us via the web, and with new types of documents emerging all the time [7] organising large collections is becoming even less-trivial than it has always been. One obvious target for research is to develop the ability to automatically categorise new documents; to tell *between* one type and another. However, with so much data, it is desirable to have further ways to subdivide categories; to make distinctions *within* types.

This paper is interested in one specific CMC-based document class, the online journal weblog, or 'blog'. This paper introduces two aspects of a larger study [12] which has looked at linguistic features of blogs both as one genre amongst many, and as capable of demonstrating variation within.

With so many host services, authors with multiple blogs, and the lack of statistics on non-English language blogs, quoting the number of blogs in existence is difficult. However, as an example of their increasing popularity, the host LiveJournal has seen a 10000% increase in registrations annually from the year 2000 to 2005.

With the emergence of so many different genres on the web [7] there is certainly interest in automatically distinguishing document types [17]. However, the fluidity of genres such as blogs and the freedom for individual expression available to authors means there is a great deal of variation within just this one type. This freedom provides the perfect opportunity for the exploration of variation due to individual differences: in the case of this work, personality traits. Just as automatic identification of text types is a desirable target, so is the automatic differentiation of author types.

This paper presents highlights from a larger body of work investigating the linguistic properties of, and variation within, blogs. First it describes the background to the approaches used: a unitary measure of contextuality that can be calculated for different genres of text; and a number of linguistic analyses approaches that can be related to personality; personality traits are also introduced. Secondly, the paper introduces the corpus of personal weblog text to be studied. The paper will then show how blogs are situated amongst a collection of other text genres, both CMC and non-CMC. It then reports work which shows that there are linguistic features that can be used to distinguish personality traits.

2 Background

2.1 Contextuality of language

Heylighen and Dewaele [9] explored the notion of implicitness in a text by developing a unitary measure of contextuality. They considered parts-of-speech as they related to *deixis*: that is to say POSs that generally require anchoring within the spatio-temporal context of an utterance in order to be properly interpreted; for example pronouns can generally be considered diectic, or highly contextual, while nouns are (generally) non-diectic, or less contextual. Their F-measure is defined as follows:

$$F = 0.5 * [(nounfrq + adjfrq + prepfrq + artfrq) - (pronfrq + verbfrq + advfrq + intfrq) + 100)]$$

The F-measure was use to explore data derived from multiple language and the results were consistent:

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.

spoken language scored lower than written language, meaning that the latter is less contextual; fiction is more contextual than newspapers. Of course, there are other factors which can be used to distinguish *between* genres [2, 10]. However, the F-measure has also been used specifically to investigate individual differences between writers *within* a genre, hence the adoption of this measure.

2.2 Personality traits

This work explores personality from the perspective of Costa and McCrae's five-factor model [6]. Each factor gives a continuous dimension for personality scoring. The factors, defined here by their facets [11], are: *Neuroticism* (anxiety, angry hostility, depression, self-consciousness, impulsiveness, and vulnerability); *Extraversion* (warmth, gregariousness, assertiveness, activity, excitement-seeking, and positive emotion); *Openness to experience* (fantasy, aesthetics, feelings, actions, ideas, and values); *Agreeableness* (trust, straightforwardness, altruism, compliance, modesty, and tender-mindedness); and *Conscientiousness* (competence, order, dutifulness, achievement striving, self-discipline, and deliberation)

2.3 Linguistic features

The first approaches employed were content analyses, using categorised dictionaries of words. The Linguistic Inquiry and Word Count (LIWC; [14]) is a collection of psychologically-derived, human-constructed words categories. For example, the *Social Processes* category contains words such as 'talk', 'us' and 'friend', whilst *Causation* words include 'because', 'hence' and 'effect'. The LIWC has been used previously to study both language and personality [15] and the language of blogs [4]. The MRC psycholinguistic database [5, 18] was originally developed as a resource for researchers, but was applied in this context following Gill [8]. It contains data about, for example, the concreteness and standard age of acquisition of words. In addition to these top-down features, bottom-up features are included in the form of POS counts from calculating the F-measure (as described in section 2.1) and distinctive word collocations — bigrams and trigrams that proved to be significantly used by one personality sub-group over another.

3 The weblog corpus

3.1 Construction

A corpus of blog text has been gathered [12]. Participants were recruited directly via e-mail to suitable candidates, and indirectly by word-of-mouth: many participants wrote about the study in their blogs. Participants were first required to answer sociobiographic and personality questionnaires. The personality instrument was specifically designed for online completion [3]. Participants rate themselves on 41-items using a 5-point Likert scale, providing scores for the traits described in section 2.2.

After completing this stage, participants were requested to submit one month's worth of prior weblog postings. This month was pre-specified so as to reduce the effects of an individual choosing their 'best' or 'preferred' month. Raw submissions were marked-up using XML, distinguishing post types such as purely personal, commentary reporting of external matters, or direct posting of internet memes such as quizzes. The corpus consisted of 71 participants (47 females, 24 males; average ages 27.8 and 29.4, respectively) and only the text marked as 'personal' from each weblog, approximately 410,000 words. To eliminate undue influence of particularly verbose individuals, the size of each weblog file was truncated at the mean word count plus 2 standard deviations.

3.2 Personality distribution

A common misconception regarding the personality of bloggers is that they are narcissistic exhibitionists; i.e. Extraverted. This assumption appears to be incorrect, since plotting the distribution of Extraversion scores (figure 1) reveals a relatively normal distribution. However, when Openness scores are plotted (figure 2) there is a significant bias in the sample. It is conceivable that blogger are more Open than average; or perhaps there is response bias. However, without a comparison sample of matched non-bloggers, one cannot say for certain. Due to the statistical complications this creates, Openness is not discussed further in this paper.

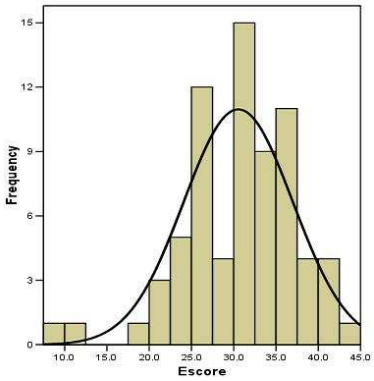


Figure 1: Distribution of Extraversion scores

4 Between Genres

Looking at blogs as a whole we compare them to a range of genres selected from the British National Corpus (BNC). The BNC consists of over 4000 files, containing over 100 million words of both spoken and written English. Calculating the F-score of a selection of genres from the BNC allows us to place blogs on a scale.

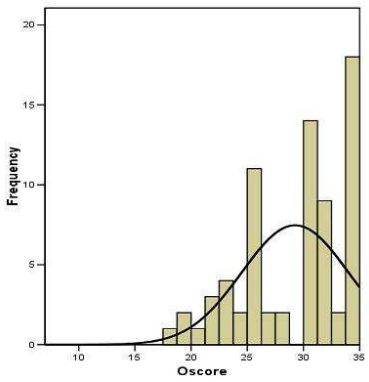


Figure 2: Distribution of Openness scores

4.1 Method

Using Lee's BNC World Edition Index[1] (2001), 17 genres were selected from the BNC. These included both spoken ($n = 4$) and written ($n = 13$) material. Only files dating from 1985 to 1994 and (for speech) only files with a single speaker were included. Altogether there were 837 files comprised of 23 million words. The original release of the BNC comes pre-tagged, and these tags are algorithmically reduced to the set needed for calculating the F-score of each file. These scores are then averaged to give the F-score of each genre. The F-score for the blog corpus was also computed, and in addition, that of the e-mail corpus of Gill [8].

4.2 Results

When the F-score calculations were completed, the genres ranked as in Table 1. As predicted by Heylighen and Dewaele [9], spoken genres are on the whole more contextual than written, with sermons, lectures, and unscripted speeches scoring the lowest. As expected, unscripted Speeches are more contextual than scripted, while fiction is more contextual than academic writing. Genres appear to be ordered in a plausible manner.

As one might expect, the e-mail corpus is very similar to the E-Mails taken from the BNC; proximity to Personal Letters follows from this. It can be seen that the blogs are scored as being significantly less contextual than the e-mails (t=3.54, DF=174, $p<.001$), scoring similarly to School-level essays.

4.3 Discussion

That blogs are less contextual than e-mail can be explained by considering some of the situational factors involved in deixis. Heylighen and Dewaele describe four categories: the *persons* involved, the *space* of the communication, the *time*, and the prior *discourse*. The e-mail corpus consists of two emails per subject, written to a good friend. Blogs however, as a property of being published online, can be read by anyone; hence, to at least some degree, they are written with such readers in mind. Bloggers therefore cannot assume as large a shared context, if any, with their readers as writers of e-mails composed for friends.

Not knowing the reader means the writer can assume less about their knowledge of any places, or *spaces* that are discussed. Similarly, since one cannot know when a reader will encounter their blog, or if they have read it previously, the writer can assume less about the *time* and *discourse* contexts.

It appears then that the F-measure, a measure of contextuality of language, is a reasonable method for distinguishing *between* genres. In fact, the ordering on genres is very similar to that found by Biber [2] when ranking via his involved/informational factor. The standard deviations shown in table 1, however, show that there is greater variance within some genres, although there does not appear to be a clear pattern. This is perhaps an effect of the number of files that each of the genres consists of (ranging from 3 to 374) and the level of individual variance within (cf. [13] for discussion of F-score variation due to personality within blogs).

Genre	Ave F	*(SD)*
Sermons	42.4	*(2.6)*
Lectures on Social Science	44.3	*(2.8)*
Unscripted Speeches	44.4	*(4.4)*
Fiction Prose	46.3	*(4.0)*
Personal Letters	49.7	*(3.3)*
Sports Mailing List E-Mails	50.0	*(0.6)*
E-Mail Corpus	*50.8*	*(4.0)*
Scripted Speeches	53.0	*(2.9)*
School Essay	53.2	*(2.7)*
Blog Corpus	*53.3*	*(5.1)*
Biography	56.3	*(6.4)*
Non Academic Social Science	56.9	*(6.0)*
Nat Broadsheet Social	57.5	*(3.9)*
Professional Letters	57.5	*(4.2)*
Nat Broadsheet Editorial	58.1	*(1.4)*
Nat Broadsheet Science	60.0	*(3.2)*
University Essays	60.3	*(0.6)*
Academic Social Science	60.6	*(3.3)*
Nat Broadsheet Reportage	62.2	*(1.3)*

Table 1: Average F-score (and standard deviation) of selected genres from BNC

5 Within Genre

We have so far explored a method for distinguishing between genres. We now report an exploration into the blog genre considering the personality of the author.

5.1 Method

In section 2.3 we introduced a number of linguistic features, namely the categories of the LIWC and MRC along with word n-grams. Firstly we describe the creation of the n-gram set.

Only 2/3-grams with a corpus frequency $\geq$5 were included to allow accurate log-likelihood G^2 statistics to be computed [16]. Distinct collocations are identified

[1] Available at http://clix.to/davidlee00

via a three way comparison between the high and low groups (defined as one standard deviation above and below the mean score) of each trait and a third, neutral group. This neutral group contains all those individuals who fell in the medium group for *all four traits in the study*. Hence, this approach selects features using only a *subset* of the corpus. N-gram software was used to identify and count collocations within a sub-corpus [1]. For each feature found, its frequency and relative frequency are calculated. This permits relative frequency ratios and log-likelihood comparisons to be made between High-Low, High-Neutral and Low-Neutral. Only features that prove distinctive for the H or L groups with a significance of $p < .01$ are included in the feature set.

Once all the features were identified the relative frequencies of each were computed for each individual author. These were then correlated (Pearsons r) with the personality trait scores. Any features which correlated with at least marginal significance ($p < .1$) were considered to show a relationship with the personality trait in question. This produces a set of related features (drawn from the LIWC, MRC, F-measure and n-grams) for each trait.

In order to explore just how much of a relationship these features had with personality when combined, multiple linear regression was used. For this analysis, the traits are considered the dependent variables, while the correlating features are considered independent. The results of these analyses will provide a further sub-set of features which, when combined, explain the greatest percentage of the variation within the personality scores.

5.2 Result

In mind of space considerations, the full equations resulting from the regression analyses are not included here. Table 2 shows how much of the variance is explained, by how many independent variables along with how significant the result is.

Trait	# of features	R^2	p
N score	10	**.67**	**.000**
E score	8	**.55**	**.000**
A score	8	**.65**	**.000**
C score	8	**.66**	**.000**

Table 2: Multiple regression analysis with personality scores

The third column, the R^2 value, can be seen as the percentage of variance explained by the independent variables. So it is clear that a combination of 10 linguistic features accounts for 67% of the variation in Neuroticism. Similarly, 55% of Extraversion, 65% of Agreeableness and 66% of Conscientiousness can each explained by combinations of 8 features.

5.3 Discussion

These results show that just a small number of linguistic features can account for a great deal of variance. What this shows is that there are linguistic features that can be used to differentiate between personality types. In the case of Conscientiousness for example, calculating the relative frequency of just 8 features in a text offers a reasonably reliable tool to identify high scorers from low. While these results do not translate directly into automatic classification, they are a promising start.

It is interesting to note which features proved most useful. Though exact details are not given here, it must be brought to the readers attention, that the majority of the features retained in the analyses were from the n-gram sets. In fact only 6 of the 34 features were not n-grams. N-gram frequency is trivial to compute for individual documents. This suggests that n-grams would be a reasonable base from which to begin experimentation in automated classification.

It is worth noting that the methodology here is perhaps slightly naïve. The use of the neutral group in identifying the distinct collocations was intended to minimise over-fitting in the correlation and regression analyses. However, it remains the case that there were only 71 subjects, and data-sparseness is likely.

6 Final words

There are many ways to separate documents. This paper has considered doing so by genre, as well as by author type. The unitary measure employed here, the F-measure, whilst perhaps not lending itself to automatic classification of individual documents, is a useful way to visualise some aspects of the differences between genres. It has proved particularly useful in highlighting the differences between the CMC genres of blogs and e-mails. In the second study reported we have shown that there are features which can be used to detect personality traits. In combination, these explain considerable levels of variation within the language used by different personality types. This suggests that it might not be such a wild idea to consider the automatic classification of text by author personality.

Acknowledgements We would like to thank Robert Dale for his invaluable comments and encouragement. The first author also acknowledges funding from the UK Economic and Social Research Council.

References

[1] Satanjeev Banerjee and Ted Pedersen. The design, implementation, and use of the ngram statistics package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, 2003.

[2] Douglas Biber. *Variation Across Speech and Writing*. Cambridge University Press, Cambridge, 1988.

[3] Tom Buchanan. Online implementation of an ipip five factor personality inventory. Available at

http://users.wmin.ac.uk/∼buchant/wwwffi/introduction.html, *accessed 07/10/06*, 2001.

[4] Michael A. Cohn, Matthias R. Mehl and James W. Pennebaker. Linguistic markers of psychological change surrounding september 11. *Psychological Science*, Volume 15, pages 687–693, 2004.

[5] M. Coltheart. The mrc psycholinguistic database. *Quarterly Journal of Experimental Psychology*, Volume 33, Number A, pages 407–505, 1981.

[6] Paul T. Costa and Robert R. McCrae. *Revised NEO Personality Inventory (NEO-PI-R) and NEO Five-Factor Inventory (NEO-FFI): Professional Manual*. Odessa, FL: Psychological Assessment Resources, 1992.

[7] Kevin Crowston and Marie Williams. Reproduced and emergent genres of communication on the world wide web. *The Information Society*, Volume 16, Number 3, pages 201–216, 2000.

[8] Alastair J. Gill. *Personality and Language: The projection and perception of personality in computer-mediated communication*. Ph.D. thesis, University of Edinburgh, 2004.

[9] Francis Heylighen and Jean-Marc Dewaele. Variation in the contextuality of language: an empirical measure. *Foundations of Science*, Volume 7, pages 293–340, 2002.

[10] Max Louwerse, Philip M. McCarthy, Danielle S. McNamara and Arthur C. Graesser. Variation in language and cohesion across written and spoken registers. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society*, pages 1035–1040, Hillsdale, NJ, 2004. LEA.

[11] Gerald Matthews, Ian J. Deary and Martha C. Whiteman. *Personality Traits*. Cambridge University Press, Cambridge, 2nd edition, 2003.

[12] Scott Nowson. *The Language of Weblogs: A study of genre and individual differences*. Ph.D. thesis, University of Edinburgh, 2006.

[13] Scott Nowson, Jon Oberlander and Alastair J. Gill. Weblogs, genres and individual differences. In *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, pages 1666–1671, Hillsdale, NJ, 2005. Lawrence Erlbaum Associates.

[14] James W. Pennebaker and Martha E. Francis. *Linguistic Inquiry and Word Count: LIWC*. Erlbaum Publishers, 1999.

[15] James W. Pennebaker and Laura King. Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, Volume 77, pages 1296–1312, 1999.

[16] Paul Rayson. *Wmatrix: A statistical method and software tool for linguistic analysis through corpus comparison*. Ph.D. thesis, Lancaster University, 2003.

[17] Maria Santini. Clustering web pages to identify emerging textual patterns. RECITAL 2005, Dourdan, 2005.

[18] Michael Wilson. MRC psycholinguistic database: Machine usable dictionary. Technical report, Oxford Text Archive, Oxford, 1987.

Dual Interactive Information Retrieval

Vitaliy Vitsentiy
Queensland University of Technology
National ICT Australia

v.vitsentiy@qut.edu.au

Abstract *A new task in Interactive Information Retrieval (IIR) is considered – optimization of information retrieval taking into account impact on quality of interaction with the user. Dual IIR is defined.*

Keywords dual interactive information retrieval, multistage stochastic programming.

1 Introduction

Information Retrieval Systems (IRSs) are used for search of indistinct information. The user has some information need, which they translate into some query language that usually can not describe the information need precisely. From the point of view of the IRS there is an uncertainty what is searched by the user. This is caused by: 1) information need is not precise; 2) information need is not represented precise enough by the query; and 3) information need changes. So, IRSs make decisions about which documents to select based on the user query in situation where there is not enough information for unambiguous decisions. IRS usually retrieves not one document but a portion of documents, among which the user may find the necessary document. Often information need is not satisfied at once and the search process goes on (see Figure 1). Thus Interactive Information Retrieval (IIR) may be considered as a problem of multistage decision making under uncertainty. Such problems are usually solved by stochastic programming methods [2].

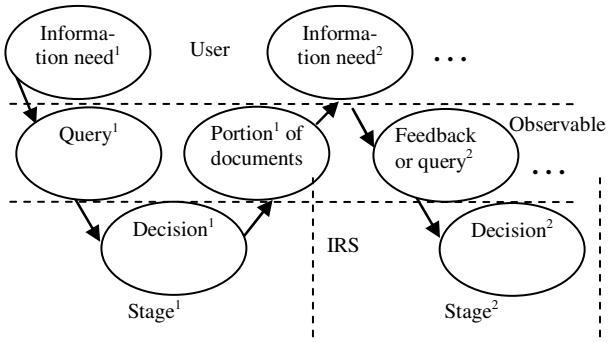


Figure 1: Dynamics of information search.

2 Definition

Dual IIR (DIIR) is information retrieval with feedback after each stage of retrieval when at selection of documents the IRS takes into account not only their relevance but also possible effect of the retrieved portion of documents on user feedback for optimization of the whole user search session. The term "dual" is taken by analogy with "dual control" from adaptive control theory. The dual features are: 1) direction: DIIR System retrieves more relevant documents in the current portion; 2) probing: DIIR System encourages receiving better user feedback. The need for duality is caused: 1) user information need is uncertain and the uncertainty should be reduced; 2) the feedback is evaluative, not instructional and therefore it does not allow determining whether there are more relevant documents among the not retrieved documents than the retrieved documents. The expediency of the development of the methods is based on the hypothesis: if user feedback is encouraged to be better especially on early stages of search, even by retrieving documents with somewhat lower relevance, then IRS using feedback can estimate relevance of documents more precisely and this leads to better relevance of all documents in the whole user search session.

Retrieving documents in DIIR is not simple selection of documents with the largest relevance but is an optimization problem. Because the optimal value is determined by: 1) probing may lead to lower relevance of documents in the current portion but a better feedback will allow to determine relevance better and retrieve more relevant documents on the next stages; 2) an increase of quantity of retrieved similar documents may lead not only to increase of relevance of the current portion but also to decrease of feedback quality and accordingly to decrease of relevance on the next stages.

The problem of DIIR includes into the IR decision making also the task similar to active learning [3]. So, DIIR is a problem of adaptive dual control [1], where the problem of optimal balance between control and estimation is studied. A mathematic programming model for decision making in DIIR can be found in [4].

3 Conclusion

DIIR is defined. A positive effect from usage of DIIR should be expected in the situation of vague user information need, long search interaction process, broad semantic range of search.

References

[1] A.A. Fel'dbaum, "Dual Control Theory. I-IV" *Automation Remote Control*, 21, 22, pp. 874-880, 1033-1039, 1-12, 109-121, 1960-1961.

[2] *Stochastic Programming*, ed. by Ruszczynski A., and Shapiro A., Elsevier, 2003.

[3] S. Tong, *Active learning: theory and applications*, PhD thesis, Stanford University, 2001.

[4] V. Vitsentiy, "A decision making model for dual interactive information retrieval", *International Conference on Systems, Computing Sciences and Software Engineering 2006*, in press.

Enhanced web-based translation extraction for English-Chinese CLIR

Chengye Lu Yue Xu Shlomo Geva
School of Software Engineering and Data Communications
Queensland University of Technology
Brisbane, QLD 4001, Australia
{c.lu,yue.xu,s.geva}@qut.edu.au

ABSTRACT

Dictionary based translation is a traditional approach in use by cross-language information retrieval systems. However, significant performance degradation is often observed when queries contain words that do not appear in the dictionary. This is called the Out of Vocabulary (OOV) problem. The common methods for translation selection for web-based translation always rely on word frequency calculation but the results are not always satisfactory. Our experiments show marked improvement in translation accuracy over other commonly used approaches.

1. INTRODUCTION

Dictionary based translation has often been used in cross-language information retrieval because bilingual dictionaries are widely available and dictionary approaches are easy to implement. This approach shows high efficiency in term and phrase translation, however, translation disambiguation and the out of vocabulary (OOV) problem challenge cross-language information retrieval systems. Translation disambiguation refers to finding the most appropriate translation from several choices in the dictionary. The OOV problem refers to the situation where translations of some words cannot be found in the dictionary at all. Even in the best of dictionaries this is to be expected of course. Very often the OOV terms are proper names or newly created words that carry the most information of the query. When it is missing in the translated query, it is most likely that the user will practically be unable to find any relevant documents at all.

2. PROPOSED APPROACH

Our approach is similar to the previous works[1][2][3] in terms of the web based translation approach which tries to find the OOV term's translation through web search engine. However, our approach differs in term ranking and selection strategy. The aim of our approach is to find the most appropriate translation from the word list regardless the term frequency.

The basic idea of our approach is to combine the translation disambiguation technology and the web-based translation extraction technology together. The web-based translation extraction process usually returns a list of words. As those words are all extracted from the results returned by the web search engine, it is reasonable to assume that those words are relevant to the English terms that were submitted to the web search engine. If we assume all those words are translations of the English terms, we can apply the translation disambiguation technique to select the most appropriate word as the translation of the English terms..

2.1 Results

Table 1 below gives the results from four runs.

Table 1 Retrieval performance

	Average precision	Percentage of Mono
Mono	0. 3526	100%
Ignore OOV	0.1290	36.5%
Previous	0.2302	65.3%
Propose	0.2576	73.1%

It is clearly that when using our proposed approach, we have highest retrieval performance. This result indicates that our translation approach has the highest effective. The precision of our approach is 174% comparing to the case of not processing OOV terms and it is 120% comparing to the case of the simulation of previous approaches.

3. CONCLUSION

In this paper, we have described an approach to tackling the OOV problem in English-Chinese information retrieval. By using web translation extraction based on co-occurrence model, the overall performance can boost to almost 174% comparing to the case of not processing OOV terms. 120% comparing to the simulation of previous approaches. This is a marked improvement in translation accuracy over other commonly used approaches.

4. Reference

[1] Cheng, P.-J., J.-W. Teng, et al. (2004). *Cross-language information retrieval: Translating unknown queries with web corpora for cross-language information retrieval.* Proceedings of the 27th annual international conference on Research and development in information retrieval.

[2] Gao, J., J.-Y. Nie, et al. (2001). *Improving query translation for cross-language information retrieval using statistical models*. SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, United States, ACM Press.

[3] Zhang, Y., and Vines, P (2004). *Using the web for automated translation extraction in cross-language information retrieval*. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. Sheffield, United Kingdom, ACM Press.

Examining the Pseudo-Standard Web Search Engine Results Page

Andrew Turpin
Dept. of Computer Science & IT
RMIT University
Melbourne, Australia
aht@cs.rmit.edu.au

Falk Scholer
Dept. of Computer Science & IT
RMIT University
Melbourne, Australia
fscholer@cs.rmit.edu.au

Bodo Billerbeck
Sensis Pty Ltd
Melbourne, Australia
Bodo.VonBillerbeck@sensis.com.au

Larry A. Abel
Dept. Optometry and Vision Science
University of Melbourne
Melbourne, Australia
label@unimelb.edu.au

Abstract Nearly every web search engine presents its results in an identical format: a ranked list of web page summaries. Each summary comprises a title; some sentence fragments usually containing words used in the query; and URL information about the page. In this study we present data from our pilot experiments with eye tracking equipment to examine how users interact with this standard list of results as presented by the Australian `sensis.com.au` web search service. In particular, we observe: different behaviours for navigational and informational queries; that users generally scan the list top to bottom; and that eyes rarely wander from the left of the page. We also attempt to correlate the number of bold words (query words) in a summary with the amount of time spent reading the summary. Unfortunately there is no substantial correlation, and so studies relying heavily on this assumption in the literature should be treated with caution.

Keywords web search engine, eye tracking, web page summaries

1 Introduction

All major Internet search engines such as Google, Yahoo!, MSN Search, and Sensis present answers to queries in a similar format, as typified by the screenshot in Figure 1. The top section of the answer screen contains some searching options and the query in an editable box, with the majority of the screen filled with a list of *summaries* of web pages. Each of these summaries is composed of four parts:

1. the page title, which is extracted from the HTML of the page;
2. a query-biased extract of the page, which is typically two to three sentence fragments that contain the query words (highlighted in bold);
3. the URL of the page; and
4. some information about the page, for example a link to its cached version, or more pages from the same domain

The bottom of the screen typically contains links to more pages of summaries, and links relevant to the search engine; and the right of the screen typically contains advertisements.

Given that the format of the results page is so ubiquitous, it has received little attention in the scientific literature. In particular, investigations of what people actually look at on the screen in relation to their searching behaviour remains unpublished, apart from some work that we summarise in the next section. Presumably search engine companies have invested a large amount of resources into studying the effectiveness of their results pages, but the results of these studies are not in the public domain.

While the main aim of many eye tracking studies related to web search is to improve methods for offline evaluation of search engines, in this paper we focus on the behaviour of the users as they read the results page. In particular, we report that users with some web searching experience look at URLs in the results page when performing a navigational type task, and generally do not read the page snippets. When performing an informational type search, the snippets are heavily read. We also attempt to correlate the number of bold/query words in a summary with the amount of time spent reading a summary, as has been suggested in the literature [1], but find no substantial correlation.

2 Related Work

The study of eye movements as a reflection of cognitive processes have been investigated in the field of psychology for over 50 years, with many studies supporting the view that shifts in viewer attention are reflected by changes in the point of visual fixation [5]. The advent of non-intrusive eye tracking technology has enabled researchers to explore the usefulness of using eye move-

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.

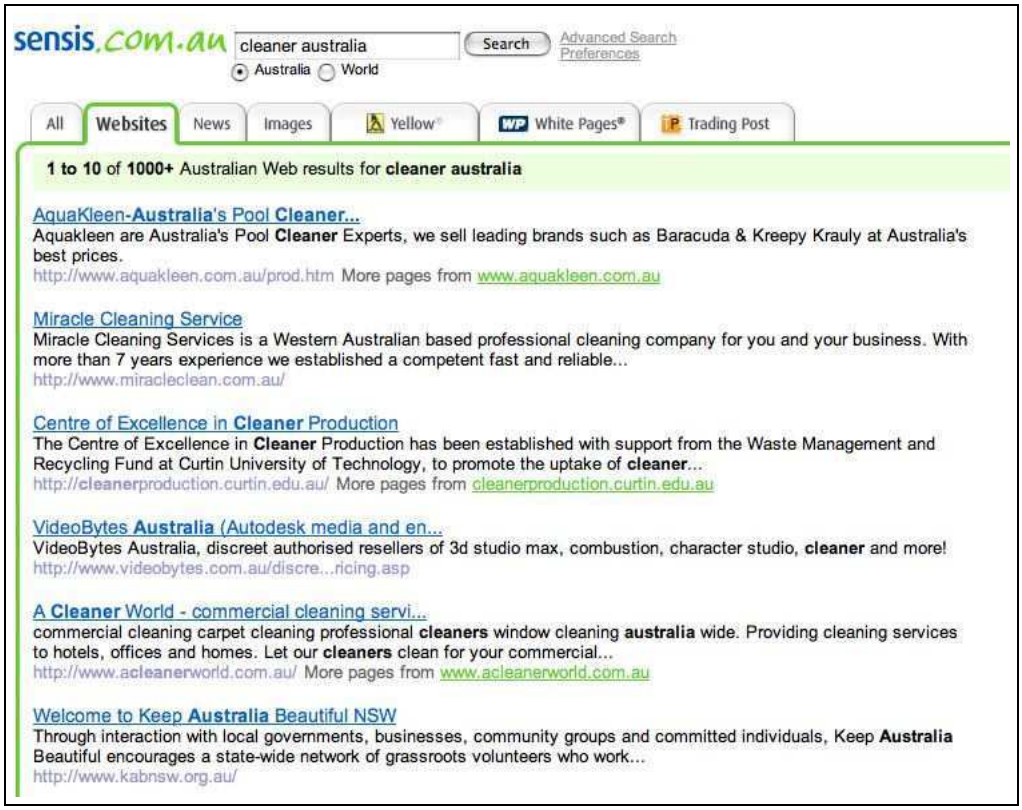


Figure 1: A screenshot of a results page of the style that is typical of most major web search engines.

ments as evidence for changes in attention in a variety of information system domains, such as the design of user interfaces, and to analyse the viewing patterns of web users [3, 7, 9].

The use of eye tracking to analyse the behaviour of online searchers was investigated by Granka [2], who conducted a series of experiments with 29 subjects. Participants carried out a series of search tasks, including informational searches (where the user is seeking to learn about a topic, for which there may be several relevant answer resources) and navigational searches (where the user is looking for a single named resource). Her results demonstrated that user search behaviour is influenced by various factors, including task type (more document summaries in a result list are viewed for informational searches than for navigational searches) and task difficulty (more summaries below a selected resource in a result list are viewed for harder search tasks, than for easier search tasks). Subject variables, such as gender, were also found to have an effect. Eye tracking analysis has also been used to demonstrate that users tend to read items in a search results list sequentially, spending significantly more time viewing the first items, and that even when users skip some items and click on an answer further down in the results list, they will generally have spent some time reading the abstracts of items that were ranked more highly [4].

In the information retrieval domain, the key challenge is to present users with resources that are relevant to an information need. As such, an interesting question to consider is whether features derived from eye tracking systems can be used to infer the relevance of items in an answer list. Salojärvi et al. [10] collected data from users viewing titles of documents with known relevance. Their results show that a discriminative Hidden Markov Model can be trained to infer relevance more effectively when using features derived from eye tracking, compared to a system that does not incorporate eye tracking data. In subsequent work, Puolamäki et al. [8] explore different statistical models to combine implicit feedback from eye movements with collaborative filtering. Their results show that more complex mixtures models are more effective than simple linear models at making relevance predictions for users of web search engine results. While these findings are promising, it is still unclear to what extent these benefits would translate into gains for users of a live search system.

Click-through data — recording those items in a list of search results that a user actually clicks on and views — has been of interest as an indicator of relevance. Trends from the eye movements of subjects as they read a search results page have been applied to the problem of validating the use of click-through data as implicit relevance judgements. Based on the positions of viewed resources in a search results list, Joachims et al. [6] investigated sources of bias in using click-through data

as an indicator of relevance. Their analysis indicated that click-through items are subject to trust bias (users tend to trust a search engine, and so are more likely to click on the first item in an answer list, even when it is not relevant), and quality bias (when the overall quality of the answer list decreases, users will view less relevant answers). They therefore conclude that click-through information should be used as relative, rather than absolute, evidence for relevance.

In recent work on user interaction models, Agichtein et al. [1] incorporate information about which sections of summaries users choose to examine: the title, snippet, or URL. However, instead of using explicit eye-tracking data, Agichtein et al. model this user behaviour by considering the extent of overlap between words that occur in the query and summary (the bold words). We investigate how closely these content-based features approximate actual user behaviours.

3 Methods

Nine users (seven male, two female) were asked to find the answer to the ten topics in the order shown in Table 1 using the `sensis.com.au` search engine. URLs selected by the users (click-through data) were recorded using a proxy between the user and the Sensis search engine. Before each topic, the user was asked to rate their prior knowledge on the six point scale shown in Table 2. All users were experienced with search engine technology — predominately postgraduate students — making use of search engines at least once a day prior to the experiment.

The hardware used was a standard PC running Microsoft Windows XP and Internet Explorer, but the monitor was equipped with the Tobii 1750 eye tracker (Tobii Technology, `www.tobii.com`), which makes use of infra-red reflections from the eye to monitor eye movements. The Tobii software, Clearview 2.0, was used to collect the eye-tracking data which consisted of the x-y co-ordinates of any fixations of gaze; the duration of any gaze in milliseconds; and timestamps for all events, button clicks, and URLs selected. The software also saves viewed pages as images.

In order to determine at which part of a web page a fixation took place (title, snippet, or URL), the x-y location data in the Clearview log file must be located on the corresponding web page image. This is non-trivial as the x-y location of a fixation is the average eye position over a short period of time (about 250 ms), and so may not actually appear "on top of" a feature in the image file. Hence we wrote a simple image processing program to perform a radial search, spiraling out from the given x-y location until a non-white pixel is encountered. The same program also segmented the web page image into summaries so that a rank equal to the rank of the summary on the page could be assigned to each gaze fixation. Not only did this software have the effect of assigning a feature and rank to each gaze fixation, it also

Informational	
1	Name the first female member of the Australian Federal Parliament
2	Name a football team/club that plays in the Northern Territory Football League.
3	What is the daily circulation of the Brisbane Courier Mail newspaper?
4	Name two Australian uranium mines.
5	With what percentage was the referendum on an Australian republic defeated in 1999?
Navigational	
6	Find the home page of CSIRO
7	Find the home page of Lion Nathan Limited
8	Find the home page of University of New England
9	Find the home page of Coles supermarkets
10	Find the home page of Federation Square in Melbourne

Table 1: Topics used in this study.

1	I do not understand the question
2	Part or all of the question makes little sense to me
3	The question makes sense, but I could not begin to guess an answer
4	I could make a poor guess at an answer
5	I could make a good guess at an answer
6	I know the answer

Table 2: Scale used to asses user's confidence in a topic prior to searching.

corrected some systematic errors in the gaze tracking due to poor calibration, and some peculiarities related to the web page images. In particular, a line of fixations left-to-right often appeared above a line of text in the image file, but clearly the user was reading that line.

Once alignment had occurred, our first analysis task was to attempt to replicate results from the Joachims et al. study [6]. In their study, the number of fixations at each rank, the number of fixations at given ranks relative to the rank of clicks, and the order of fixations on a page were all reported.

Our second task was to then examine the correlations between task type (navigational or informational) and the parts of summaries examined.

We also investigated the relationship between eye movements over parts of summaries and previously proposed features intended to approximate this behaviour, namely the overlap between query terms and terms in titles, URLs and summary parts of answer items in a search results list.

4 Results

Figure 2 shows the proportion of pages where a summary was viewed at a particular rank, and the proportion of pages where a summary was selected at a par-

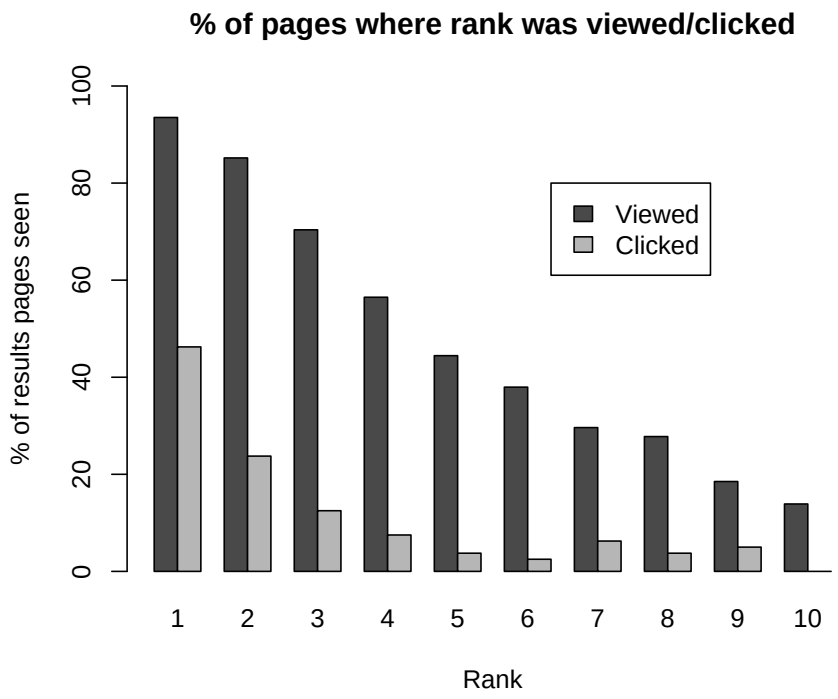


Figure 2: Proportion of pages where the summary at the rank indicated on the x-axis was either viewed or clicked.

ticular rank. In this, and all subsequent results on fixations, we only include the fixations on a page up to the first click that leads to another page. We do not include fixations on the page that may be the result of a second visit to the results page, for example via the Back button on the browser.

Consistent with previous work [6], there is a strong bias towards reading the highly ranked items, with items ranked further down the list scarcely receiving attention. Figure 3, which shows histograms of the X and Y co-ordinates of fixations further confirms that users spend most of their time looking at the top of the results page. The top histogram (for the X co-ordinates) shows a heavy bias towards fixations on the left of the page, with over 60% of fixations occuring in the left 20% of the screen for all topics. Similarly, the right histogram (for the Y co-ordinates) shows a heavy bias towards the top of the pages, with 54% of fixations in the top 20% of the screen for all topics.

Returning to Figure 2, we see a slow decline in the proportion of pages where high ranks are viewed (dark bars), but a much sharper decline in the selection of high ranks (light bars). This indicates that users read further down the list before they make their first selection. Indeed, Figure 4 confirms this observation. In this figure we show a boxplot of the rank of summaries that are viewed (have at least one fixation) as an offset from the rank of the first summary selected. Boxes indicate the quartiles of the number of summaries, whiskers and dots show extreme values, and the solid black line indicates the median. For example, for all the summaries selected when in position 1 (leftmost box in the figure), 50% of the time at least summaries in position 1, 2 and 3 were read before the click. This is indicated because the box extends down to -2 on the y-axis. For summaries selected at rank 4, 50% of the time summaries at ranks 2 and 3 received the user's gaze.

It seems apparent then, that users prefer the top-left corner of the screen when it comes to reading, and that users will read one or two summaries past the summary they eventually click upon. It would seem intuitive that the summaries are read in order from top to bottom, and this is supported by data in Figure 5, which shows the median number of fixations that occur before a fixation on the indicated rank. At the top of the results list, ranks one through seven, the list is being read in order because the number of fixations prior to arriving at a rank is increasing.

The results presented so far have all been at the summary level, reporting how users gaze at summaries as a whole. Figure 6 breaks summaries into three components: title, snippet and URL, and reports the number of seconds spent fixating on each component summed over all users. For topics 1-5 (informational) the total time spent is appreciably more than for topics 6-10 (navigational). Moreover, the relative time spent reading titles and snippets is significantly higher for the informational topics. Generally, the time to read URLs was higher than for the navigational topics.

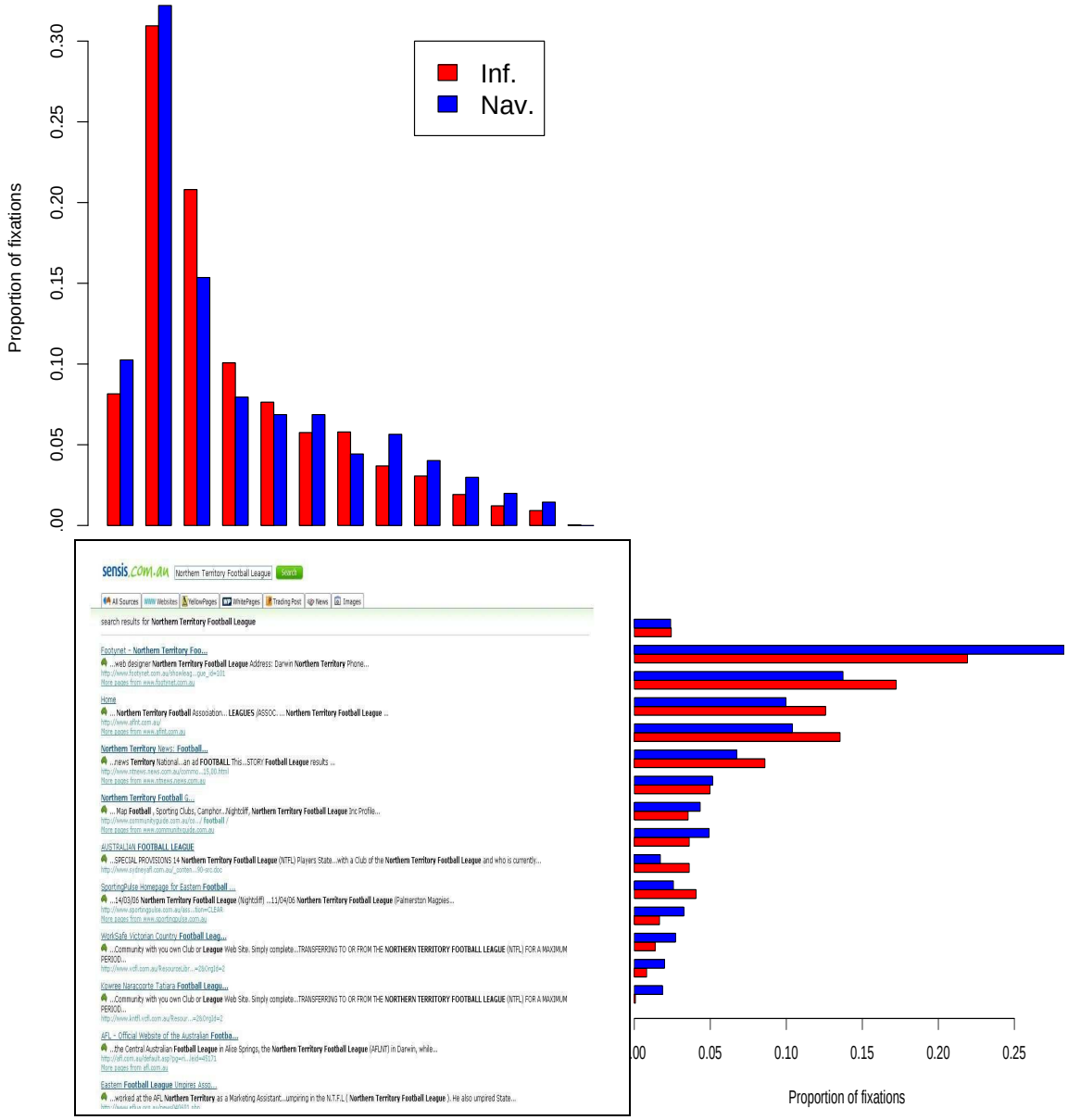


Figure 3: Histograms of the X (top) and Y (right) co-ordinates of fixations over all users. Bar heights are normalised by the total number of fixations.

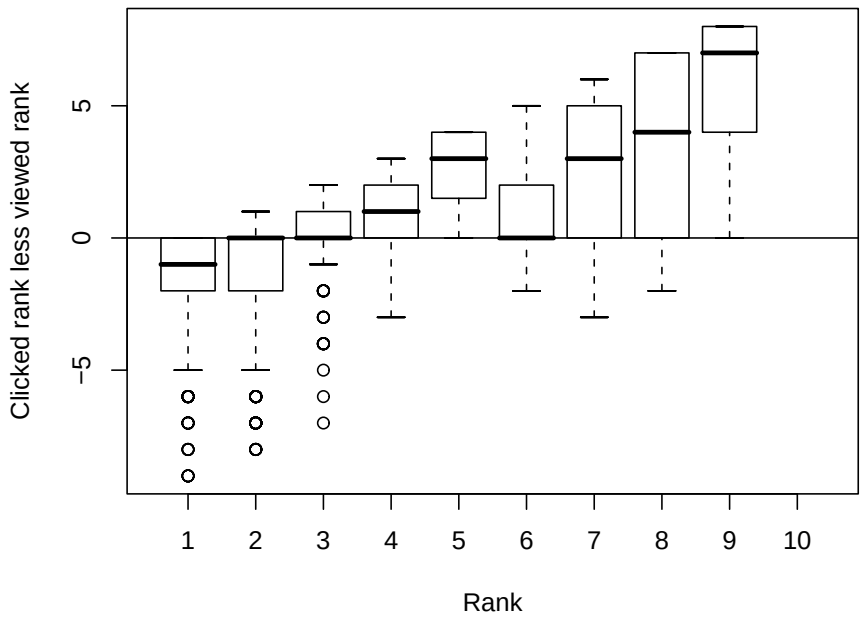


Figure 4: Rank of summaries viewed as an offset from, and grouped by, the rank at which the click took place as indicated on the x-axis.

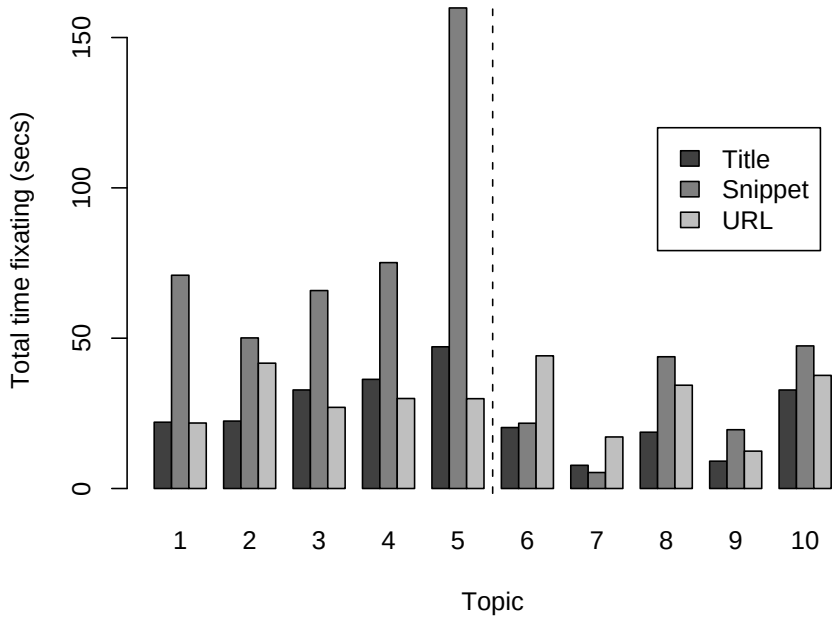


Figure 6: Number of fixations spent on each element of summaries summed across all nine users. Topics 1-5 are navigational; topics 6-10 informational.

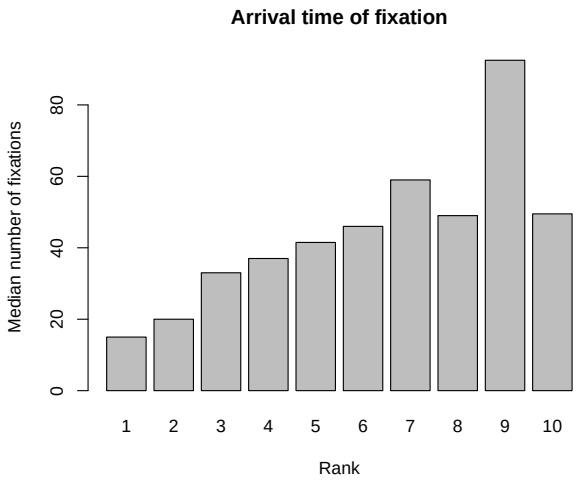


Figure 5: Median number of fixations before fixating on the summary at the rank indicated on the x-axis.

Part of page	r	p
Whole summary	0.18	< 0.001
Titles only	0.13	0.015
Snippets only	0.17	< 0.001
URLs only	0.11	0.041

Table 3: Spearman's correlation coefficient between the time spent fixating on summary components, and the number of bold words in the component.

To investigate whether a count of the overlap between terms in a query and terms in the title, snippet and URL of summaries is a suitable approximation for actual user behaviour when viewing a results page, we calculate the Spearman rank correlation between these three features and the actual time (in ms) that users fixate on these components of answer items. Table 3 shows that there are statistically significant correlations between the features and the time spent viewing those summaries, but the correlation coefficient, r, is small.

5 Conclusions and Future Work

Major web search engines present their results in a consistent way, displaying a ranked list of answer items, where each item consists of a title, a query-biased summary, and the URL of the underlying answer resource. Understanding how users view such result pages can give valuable insight into how the presentation of search results could be optimised (for example, for ease of use, or for advertising).

We have analysed the eye-movements of nine users as they engaged in a series of informational and navigational web search tasks. Our results confirm the findings that Joachims et al. [6] reported when using Google on the whole of the web, but we make use of Sensis on the Australian Web. Users view search results in order, typically reading from the top to the bottom of an answer list. Attention is mostly confined to the left-hand side of the screen, for both types of search tasks. However, the type of search task does have an effect on which components of individual answer items users focus their attention on: for informational queries, users spend relatively more time reading the query-biased summary sentences of answer items; for navigational searches, snippets are less important, with relatively more attention being given to the URL.

We also investigated the effectiveness of using simple content-based features — such as the overlap between terms in a query and in the title of an answer item — to approximate the actual duration of fixations on these answer components. In some recently reported work [1] it was assumed that there is a strong correlation between the number of query terms that appear in a summary and the amount of time a user spends reading that summary. This correlation has not been reported in any study to date. Our results indicate that there is a very weak correlation, but it would be unwise to base further studies on this assumption without further validation.

In future work, we intend to investigate techniques for combining evidence from eye-tracking data with click-through data, to examine the effectiveness of implicit indicators of relevance.

Acknowledgments This work has been supported by Sensis Pty Ltd and the Australian Research Council (Turpin).

References

[1] Eugene Agichtein, Eric Brill, Susan Dumais and Robert Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, Seattle, WA, 2006.

[2] Laura Granka. Eye-tracking analysis of user behaviour in online search. Master's thesis, Cornell University, 2004.

[3] Laura Granka, Helene Hembrooke and Geri Gay. Location location location: viewing patterns on www pages. In *ETRA '06: Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 43–43, San Diego, CA, 2006.

[4] Laura Granka, Thorsten Joachims and Geri Gay. Eye-tracking analysis of user behavior in www search. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 478–479, Sheffield, United Kingdom, 2004.

[5] Mary Hayhoe and Dana Ballard. Eye movements in natural behaviour. *Trends in Cognitive Sciences*, Volume 9, Number 4, 2005.

[6] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, Salvador, Brazil, 2005.

[7] Bing Pan, Helene Hembrooke, Geri Gay, Laura Granka, Matthew Feusner and Jill Newman. The determinants of web page viewing behavior: an eye-tracking study. In *ETRA '04: Proceedings of the 2004 symposium on Eye tracking research & applications*, pages 147–154, San Antonio, TX, 2004.

[8] Kai Puolamäki, Jarkko Salojärvi, Eerika Savia, Jaana Simola and Samuel Kaski. Combining eye movements and collaborative filtering for proactive information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 146–153, Salvador, Brazil, 2005.

[9] R. S. Rele and A. T. Duchowski. Using eye tracking to evaluate alternative search results interfaces. In *Proceedings of the Human Factors and Ergonomics Society*, Orlando, FL, 2005.

[10] Jarkko Salojärvi, Kai Puolamäki and Samuel Kaski. Implicit relevance feedback from eye movements. In Duch, Kacprzyk, Oja and Zadrozny (editors), *Artificial Neural Networks: Biological Inspirations - ICANN 2005*, pages 513–518, Warsaw, Poland, 2005.

Improving rankings in small-scale web search using click-implied descriptions

David Hawking
ICT Centre
CSIRO
ACT 2601 Australia
david.hawking@csiro.au

Tom Rowlands
ICT Centre
CSIRO
ACT 2601 Australia
tom.rowlands@csiro.au

Matt Adcock
ICT Centre
CSIRO
ACT 2601 Australia
matt.adcock@csiro.au

Abstract *When a searcher submits a query Q and clicks on document R in the corresponding result set, we may plausibly interpret the click as a vote that Q is a description of R. We call the Q and R pairing a 'click description'. Click descriptions thus derived from search engine logs can be accumulated into surrogate documents and used to boost retrieval effectiveness in a similar fashion to anchor text.*

We investigate the usefulness of click description surrogate documents in processing queries for an external web site search service for four organisations. Using the mean reciprocal rank of best answers as the measure of performance, we show that, for popular queries, click description surrogates significantly outperform both anchor text surrogates and the original proprietary rankings. The amount of click data needed to achieve a high level of retrieval performance is surprisingly small for popular queries. Thanks to terms shared between queries, click description surrogates can answer queries for which no specific click data is available. We show a 92% improvement due to this effect for a set of lengthy, less popular queries.

We also discuss issues such as spam rejection, unpopular queries, and how to combine click description scores with other evidence. We argue the potential of click descriptions in non-web applications where link and anchor text evidence is unavailable.

Keywords Information Storage and Retrieval, Content Analysis and Indexing [Indexing methods]

1 Introduction

Many search engines not only log query submissions but also record details each time a user clicks on a search result. This 'click data' has previously been exploited in a variety of ways:

1. as low cost judgments in evaluating and tuning search engine performance [13, 14, 20, 1, 2]
2. as a query-independent page popularity score, used in similar fashion to PageRank [16] or indegree [5]
3. as a query-dependent popularity score [15]
4. to infer similarities between a pair of web pages on the basis that they were both clicked on in the same "session" [23, 24]
5. to infer descriptions of clicked-on web pages [24, 9] (When a searcher submits a query Q and clicks on document R in the corresponding result set, we infer that Q is a description of R.)

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December, 2006.

We focus on the potential of the last method, click-implied descriptions, treated in similar fashion to descriptions derived from anchor text, to contribute to effective search. An attraction of click-implied descriptions is that they may be used for collections in which there are no links and no anchor text.

The present study investigates the retrieval value of surrogate documents consisting only of concatenated descriptions inferred from clicks, where the inverse rank of the nominated best answer to each of a large set of queries is used as the effectiveness measure. Results for different types of query are presented for four different types of organisations; stock exchange, government, media and bank.[1]

2 Relation to past work

Document surrogates containing both anchor text and query associations have been found to improve retrieval effectiveness. Indeed, Web search engines have long made use of anchor text to improve result quality [4]. A variety of methods of using click data to improve ranking have been both described in the literature and exploited in commercial products.

2.1 Surrogates and supplements

For retrieval purposes, a text document may be *supplemented* with additional terms derived from external sources such as metadata, anchor text and so on. In the case of document *surrogates*, the additional terms form their own document which is used instead of the original. Retrieval may be based on scoring the surrogate collection or those scores may be combined with scores

[1] Note that currently available test collections, TREC for example, are not distributed with applicable query logs and click data.

from the original collection. The following are examples of the use of surrogate or supplemented documents.

Sakai and Sparck-Jones [18] report that effectiveness in precision-oriented search is maintained when original documents are replaced by generic summaries during indexing. Craswell et al. [6] show dramatic improvements on homepage finding tasks of anchor text surrogates compared to the original documents. Scholer et al. [19] construct surrogates and supplements comprising controlled numbers of queries against which the original document ranked highly. They report increased accuracy in topic-finding searches but no benefit on a homepage finding task.

Amitay et al. [3] report equivocal benefit on the TREC-8 ad hoc task from supplementing documents using query reformulation sequences from a query log. The top k documents for the last query in a reformulation sequence are supplemented with the preceding queries in the sequence.

Hawking and Zobel [12] compare retrieval performance on a variety of query sets for surrogates comprising title-only, subject and description metadata and anchor text, in university and government site search. They use the rank of the best answer to a query as the criterion of performance. Overall, anchor text surrogates perform much better than the alternatives but the advantage is reported to be query set dependent.

2.2 Exploitation of click data

In the Chinese/English search engine WebGather, Lei et al. [15] linearly combine basic document scores with both link indegree and click frequency scores. Daily counts of distinct users who clicked are computed for (Q,P) pairs where Q is a query and P a web page. Furthermore, the count is multiplied by a factor designed to compensate for user reluctance to view more than the first page of search results.

The WebGather scheme compensates for changing user interests by combining the current day's scores with an attenuated aggregate of past days' scores. Finally, it also attempts to compensate for bias against recent documents. The authors enlisted ten users to 'mark' the system's performance and found an improvement over the baseline.

Smyth et al. [20] report the use of a similar hit-matrix in the context of a community-based metasearcher.

Joachims [13] presents a machine learning approach which adapts a search engine to a particular group of users. The author describes a method for training a retrieval function based around learning preference rules in the form 'for query Q, document D_a should be ranked above document D_b'. The author shows machine learning techniques are able to tailor a meta-search engine to a small group of users with similar interests.

Joachims et al. [14] discuss the reliability of the implicit feedback that can be derived from click data. The authors conclude that while click data is useful for relative relevance judgements it is problematic for absolute relevance judgements.

Click data was used in the past by the DirectHit search engine. Culliss (the DirectHit founder credited with the original idea) provided hints in [8] that DirectHit worked by 'monitoring' the sites users selected, boosting sites on which users dwell, penalising sites people don't select and rotating new sites in for review. Through a combination of these techniques, the system 'learns' from previous searchers.

Xue et al. [23, 24] study various ways of improving web search using an August 2003 click log for MSN Search.[2] This log includes data for approximately 63 million separate clicks.[3] The data covers 862 464 distinct queries.[4] They compare the methods and an Okapi BM25 baseline using a collection comprising only the webpages referenced in the click log. Xue et al. consider three methods in which Okapi scores are propagated to other pages based on 'co-visitation' relationships. The co-visitation similarity between two web pages is defined in terms of click frequencies:

$$CVS(d_i,d_j) = \frac{F(d_i,d_j)}{F(d_i)+F(d_j)-F(d_i,d_j)}$$

where $F(d_i)$ is the total number of clicks on d_j, regardless of query and $F(d_i,d_j)$ is the sum of clicks on d_i and d_j for queries associated with both documents.

Dmitriev et al. [9] use page 'annotations', both explicit and implicit, to improve intranet search results. They suggest that explicit annotations are expensive to produce, as they require users to produce them. On the other hand, implicit annotations, such as queries tied to pages that are clicked on in the result set for the query, while cheap to produce, are subject to users clicking on the wrong page. To mitigate these problems, they describe several other methods of extracting only the most valuable implicit annotations. Using the percentage of queries with a correct result in the top ten results as a measure, they show a significant improvement when using explicit annotations over the baseline, but no significant improvement with any of the implicit annotation schemes. The sample size used in the study, however, is quite small.

Agichtein et al. [1] explore the use of implicit feedback of many types, including click frequency and click rank. They compare the use of Okapi BM25 and neural network-based ranking methods, both with and without implicit feedback integrated as evidence and as a basis for reranking. They report significant gains with just click data and further gains with large vectors of implicit feedback. They observe that implicit feedback is particularly valuable for queries with poor original results. They do not address issues of spamming and the study is focused at all-of-web search.

[2] Then powered by Inktomi

[3] Personal communication

[4] After case-folding, stopping and stemming

In a related paper at the same conference, Agichtein et al. [2] go on to discuss dealing with 'noisy' user behaviour such as spam and clicks on irrelevant documents. They suggest that implicit features contain a background noise component which may be estimated by aggregating the behaviour of all users without regard to their query. Features where they suggest this may help include click frequency, dwell time and post-search behaviour such as clicks away from the original search page. They demonstrate this with a neural network-style system and report good results, with their best example delivering a recall of over 0.43 with a precision of over 0.67, which substantially outperformed their baseline.

Of most relevance to the present work, Xue et al. [24] compare the performance of click description surrogates ('NM - Naive Method') against two variants of the co-visitation method. They linearly combine surrogate scores with scores from the original documents. Using precision[5] and authority[6] measures for ten queries, they show that, when using all available click data, all three click-based methods perform roughly twice as well as the baseline. However, NM deteriorates more rapidly as the amount of click data is reduced.

2.3 Motivation for the present study

Most systems exploiting click data have been oriented toward whole-of-Web search, where click-spam is a potentially devastating problem. We wish to explore its applicability in small-scale enterprise contexts where spam is less of an issue and where document collections and click volumes are many orders of magnitude smaller.

The collection used in [24] is a very small subset, $\approx 5 \times 10^5$ pages, of the very large MSN Search collection, $\approx 3 \times 10^9$ pages in 2003 [21]. Consequently, link graph and anchor text information are incomplete and the effectiveness of the content-only baseline may not reflect values for the full collection. Also, the baseline is not the ranking against which the click data was generated.

In this paper, we attempt to show that even a naive method like that of [24] can be effective in webs of different scales. We evaluate with sizable query sets and we compare effectiveness relative to the rankings against which the clicks were generated. We also compare the relative value of anchor text and click-description surrogates in supporting effective retrieval for different classes of query. By using surrogates in isolation (following [6]) we hope to eliminate confounding variables. We also investigate the extent to which overlapping terms between click descriptions, query term overlap, helps or harms performance. Finally, we attempt to characterize the amount of click data required to achieve good performance.

[5]Precision at 20 documents retrieved

[6]The proportion of a pre-defined ten most authoritative pages which were returned in the top 20 results

3 Experimental method

In this study we use four crawled web corpora, each with two corresponding sets of queries. We compare five rankings: a baseline proprietary ranking and four rankings using simple Okapi BM25 scoring of surrogates: anchor text, two types of click descriptions, and document content only.

3.1 Datasets

Table 1 summarizes the document and click data used in our experiments. The data was crawled from externally facing websites. Anchor text for links pointing within the sites was available. The query and click logs were obtained from the production search facility for those sites. Clicks were recorded using a logging and redirection script. No attempt was made to hide the fact we were recording click data. In many enterprises, such a redirection script would not be required as the relevant information could be extracted from web server logs.

Our choice of corpora was constrained by the data available to us, but, fortuitously, the four organisations illustrate huge variations in crawl size and click density. Table 1 shows that the government corpus has one hundred times as much data but 430 times fewer clicks per page than the stock exchange corpus. The government collection includes hundreds of web hosts while the media collection includes fifteen and the stock exchange and bank include only one. During the time the logs were collected, approximately 23% of stock exchange pages received one or more clicks, but the comparable figure for the government collection was only 1%.

Queries submitted via advanced search interfaces were excluded to simplify analysis. Click entries in the log were lightly preprocessed to remove URL encodings ('+' and '%xx'). Operators were not removed, since they affect what is retrieved.

3.2 Test queries and judgments

Enterprise search systems are often judged (by searchers and purchasers) on the basis of their ability to rank the best answer to important queries at the top of the results list. As an example, consider the query 'Windows XP' submitted to the search facility on the Microsoft site. If the Microsoft Windows XP homepage doesn't appear at rank one, both users and site publishers consider this a search failure. Therefore, we evaluate by mean reciprocal rank of the best answer and use t-tests to check significance.

Table 2 summarizes the query sets used for evaluation purposes. For each corpus we obtained two sets of test queries and best answers.

Popular queries are the top queries ranked by frequency of querying, in some cases after excluding certain queries as explained in section 3.3.1. Judgements were made in collaboration with the relevant organisation where possible.

Sitemap queries are derived from the websites sitemap in a similar fashion to that described in

Table 1: Sizes of data sets and corresponding click logs

Collection	Pages	Clicks	Clicks/Page	Distinct pages clicked	Distinct queries clicked
stock exchange	2.2×10^4	1.9×10^5	8.6	5 038	23 427
government	2.3×10^6	4.6×10^4	0.02	22 055	17 539
media	7.6×10^5	9.3×10^4	0.13	44 647	30 379
bank	2.7×10^3	6.7×10^3	2.47	1 176	3 576

[10]—entries in the sitemap become test queries and the links become the corresponding best answer. This is a low cost evaluation method in which judgments are again made by the publishing organisation.

A peculiarity of the stock exchange site is that often three letter stock codes are used as queries. For example, seventy seven of the top one hundred queries are three letters long and the vast majority of these are codes. The best page for all of the codes is, according to the stock exchange, a CGI script with the code as a parameter. Such cases (perfectly answerable by a simple mapping) are not particularly interesting. Consequently, we tested the sixty seven most popular non-stock code queries. Click-through data with three letter queries is, however, included in the click surrogates.

3.3 Baseline rankings

The rankings against which click events were logged were generated using a proprietary retrieval system which is understood to make use of metadata, anchor text, and web measures such as link counts and URL properties. The production index was constantly updated over the time studied.

For our baseline condition, we indexed a crawl which was used in all experiments. As a result, the production rankings against which clicks were generated may differ from the baselines reported here. However, we are confident that large ranking perturbations would be relatively unusual. The baseline was generated by the proprietary software and used similar indexing and query processing parameters to the production service.

3.3.1 Complications with production baselines

Facilities provided by commercial search tools may interfere with good science in various ways. Two examples are as follows.

Table 2: The query sets used for evaluation purposes. Average lengths are given in words and include stopwords.

Collection	Test type	Queries	Ave. Length
SE	Popular	67	1.31
SE	Sitemap	491	2.98
Gov	Popular	87	1.25
Gov	Sitemap	430	4.32
Bank	Popular	49	1.63
Bank	Sitemap	256	2.73
Media	Popular	45	1.6
Media	Sitemap	35	2.31

The histogram of clicks shown in Figure 1 shows discontinuities at ranks which are multiples of ten for the stock exchange, corresponding to the default numbers of results per page. Similar discontinuities are present in the corresponding plot for the government service, but at multiples of twenty, as the result pages are longer. This apparent reluctance of users to click on the next page of results, and wait, may further reduce the chance of a low-ranked result being promoted through clicks.

Another complication is the availability of links triggered by a query but generated from a mechanism such as a look-up table maintained by the search administrator, rather than from the normal ranking mechanism. Examples of this mechanism include the targeted advertisements on major Web search engines and the "Editor's Choice" links on `search.microsoft.com`.

Regrettably, in our data, clicks on such results were not logged. If the nominated best page for one of our queries was the subject of such a mechanism, it would be unlikely to receive any click descriptions. Accordingly, we eliminated from analysis the queries which were subject to such short-cuts.

3.4 Creation and scoring of surrogates

In effect, surrogate documents are created by assembling the words into documents which take the place of the original ones, along the lines of [6]. All the surrogates of a particular type are indexed as a collection. Surrogate documents are then ranked using the familiar Okapi BM25 formula [17]. For the content-only surrogate, the settings from [17] $k_1 = 2.0$, $b = 0.75$ were used, as appropriate for normal text.

Hawking et al. [11] argue that length normalisation makes little sense with anchor text surrogates. The

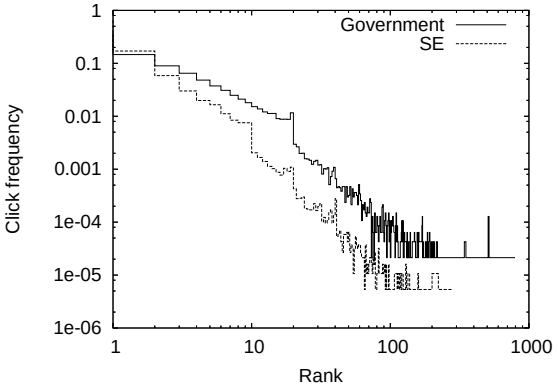


Figure 1: Reduction in click frequency as result rank increases

Table 3: Size of each collection in millions of bytes.

	Original	Anchors	Click words
bank	101.9	1.7	0.1
stock exchange	177.3	5.4	1.3
media	12 468.1	495.2	0.6
government	51 649.6	1107.5	0.7

same argument applies here as click data also provides a form of voting. Accordingly, length normalization was disabled by setting $k_1 = 2.0$, $b = 0.0$, for anchor text and click description surrogates.

Stemming and query expansion were not employed.

The sitemap pages from which the sitemap queries were derived give an obvious bias toward anchor surrogates in those cases. Consequently, the pages from which the sitemap tests were derived were removed from the index for those tests.

Four types of surrogates were studied:

Anchors —anchor text between `<a>` and `</a>` tags only from all incoming links, after following redirects.

Content only —original document, including title but excluding metadata, HTML markup, JavaScript, image tags and so on.

Click words —the surrogate for the document referenced in each pre-processed click log entry has each of the corresponding query words appended to it.

Click tokens —each distinct test case is represented by a single unique token (see Table 4). In the query log, each query that is identical to a test case is replaced with the equivalent token for the purposes of the surrogate document. This nullifies the effect of any query word overlap.

3.5 Surrogate collection sizes

As may be seen in Table 3, the sizes of the click description and anchor text surrogate collections tend to be very much smaller than the original. The difference is dramatic in the case of government where the click words surrogate corpus is around 0.001% of the size of the original.

3.6 Word overlap

Both in the case of anchor text and of click descriptions, it is possible that changes to rankings may arise from the sharing of words between different descriptions. For example, if clicks for the query 'oil corp' and clicks for the query 'stock price' hit the same document, then the word-based click description surrogate for that document contains a full match to the query 'oil corp stock price' even if no clicks have been recorded for that query. This may, or may not, be useful.

Table 4: Hypothetical example queries with example tokens

Token	Query
q1	abcd airlines
q2	oil corp
q3	wxyz corporation
q4	abcd

We investigate the effect of overlap in click descriptions by comparing effectiveness differences between the click words and click tokens surrogate collections listed above. Test queries submitted to the token collection are, of course, expressed as the appropriate tokens.

4 Experiments

In this section we describe the aims and conditions of each experiment and report results.

4.1 Experiment 1—Click effectiveness

The aims of this experiment are as follows:

- to investigate whether rankings based on click surrogates are capable of improving on the original baseline
- to compare the performance of Okapi BM25 rankings over content, anchor text and click surrogate collections
- to confirm whether patterns of results are the same on four corpora of very different sizes and on query sets devised in very different ways

Each set of queries are run against content-only, anchor text and click words corpora. The results are shown in Figures 2 and 3. We evaluate by mean reciprocal rank of best answer, and use t-tests to check significance. Several key observations may be made:

- For the bank, government and stock exchange corpora, the click desciptions ranking significantly outperforms the baseline for popular queries ($p \leq 0.01$)
- For all corpora, the click descriptions ranking is significantly outperformed by the baseline for the sitemap query sets. ($p < 0.01$)
- For the bank, government and stock exchange corpora, the click words surrogates are significantly more useful than anchors when processing popular queries. ($p < 0.01$)
- For all collections other than stock exchange, the click words surrogates are significantly more useful than content when processing popular queries. ($p < 0.01$)
- The apparent advantage to click words over content for popular queries, in the case of the stock exchange corpus, is not significant. ($p > 0.05$)

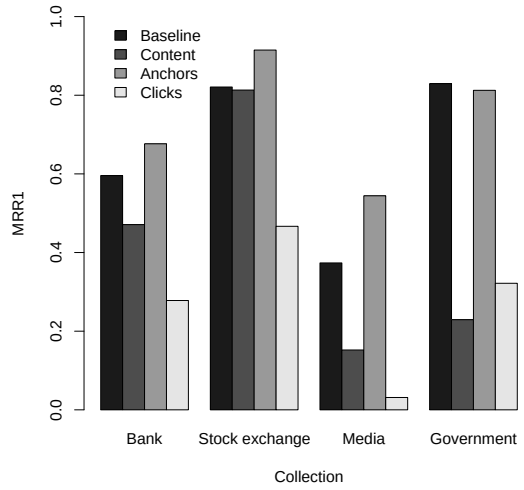


Figure 2: Sitemap tests: $p < 0.01$ except media content

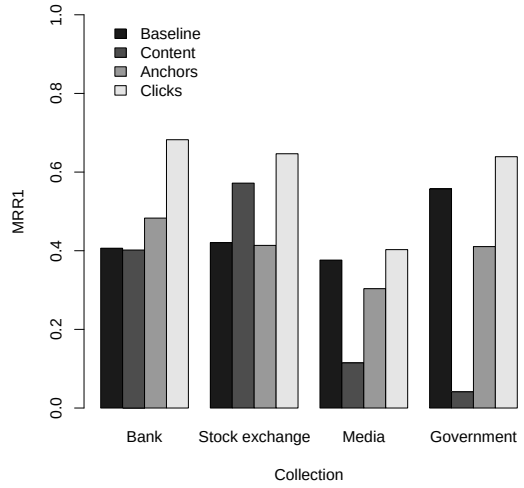


Figure 3: Popular tests: $p < 0.01$ except stock exchange content where $p < 0.09$ and media baseline and anchortext where $p > 0.1$

- In the case of the sitemap queries, click descriptions are significantly less useful than anchors on all four collections. ($p < 0.01$)

- Click descriptions in sitemap tests are significantly more useful than content only for government but significantly less useful in the other cases. ($p < 0.01$) There is no significant difference in the case of media ($p > 0.05$).

- There is a very large difference in performance for the content only surrogates, both in absolute terms and relative to the other surrogates. Performance is much higher on the smaller collections.

Table 5: Results from Experiment 2. The difference in the second row is significant. ($p < 0.01$)

Query Set	Words	Tokens
Popular	0.576	0.554
Sitemap	0.457	0.237

4.1.1 Discussion

It seems likely that the entries in a sitemap would tend to use the same language as the rest of the site, leading to higher performance for anchors and content only surrogates for sitemap based tests. On the other hand, familiarity with official nomenclature is likely to be imperfect among site visitors, tending to lead to queries (and clicks) for short queries, such as 'health', rather than to 'ministry of health and ageing'. Table 2 shows that the average length of sitemap queries is greater than popular queries; more than twice as long for stock exchange and government. The result of this may be that sitemap based queries are more specific, submitted less often and less capable of deriving benefit from click evidence.

4.2 Experiment 2—Query word overlap

The aim of this experiment is to determine the extent of harm or benefit due to word overlap across queries.

Two query sets are run against the click words and click tokens surrogate collections for the stock exchange corpus. Results are shown in Table 5. For the popular queries, there is no significant difference ($p > 0.05$, Wilcoxon signed rank significance test) between the scores, indicating that query word overlap is not important. By contrast, for the sitemap queries, query word overlap increases the MRR1 score by 92%, presumably because the exact sitemap queries are rarely submitted.

4.3 Experiment 3—Quantity of click data

The aim of this experiment was to investigate the relationship between performance on test sets and the amount of click data available. We did this by building click words corpora from randomly chosen samples of the click data.

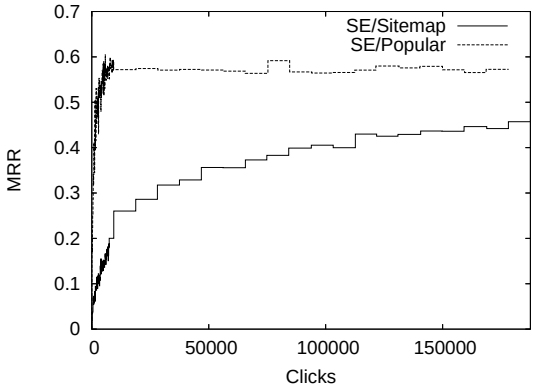


Figure 4: This figure shows the effect of varying the number of clicks used to assemble the surrogate document.

Each point in Figure 4 represents the average of two samples of approximately the same size. As may be seen, performance increases with sample size, approaching an asymptotic limit. As few as 4000 clicks are sufficient on the popular stock exchange queries to achieve an MRR score of 0.5.

The sitemap queries also start increasing rapidly but slow, approaching the final MRR less far less rapidly than the popular queries. It takes over 100 000 clicks before the sitemap queries show an MRR of 0.4 and they never reach 0.5.

5 General discussion

Our results show that click description surrogates achieve best results for popular queries. This is no doubt largely due to the larger amount of click data available for those queries, but it is a plausible supplementary hypothesis that the very short nature (average less than 1.5 words) of the popular queries makes it hard for other ranking schemes to reliably identify the best answer.

A major problem with ranking based on clicks is the potential for "fraud". By clicking repeatedly, a user can bias the ranking to favour a result. This method of artificially up-weighting results is believed by some to have lead to the demise of the DirectHit search engine [20], but we expect query dependent usage to be less susceptible than query independent popularity counts. An analogous ranking inflation technique for anchor text is "Googlebombing" [22].

Inside an enterprise, there would be no financial incentive to dishonestly manipulate rankings by clicking. Public facing enterprise websites similarly offer no incentive to manipulative clickers. Unfortunately, public websites such as a stock exchange or media site are likely to be among the exceptions to this rule.

Regardless of incentive, there may be techniques which can be used to counteract artificial clicks without excessively damaging result quality. Each click is recorded with a time, source IP address, referrer URL, query, destination and so on. It may be possible to use regularity of clicking and source address filtering with heuristics to filter out fraudulent clicks. Additionally, similarly to Web Gather [15], cookies could be used to limit the number of clicks recorded per user. Further, an asymptotic ranking function could take into account 'over-clicked' documents. The use of background click frequency, as discussed in [2] may also offer some immunity to click spam. This area of adversarial IR seems worthy of further study.

It may be possible to compensate for the trust bias discussed in [14]. Some initial experiments have been conducted by gradually downweighting clicks made on results ranked in the top ten. Initial results are equivocal. Further investigation is necessary.

Scholer et al. [19] introduced mechanisms to limit the size of query association surrogates in order to stay within defined storage limits. Here, the size of the click surrogates arising from the query log data are small relative to anchors surrogates and very tiny compared to the original data. Click surrogate sizes increase with increasing query volume, but size can easily be controlled by sampling or using a temporal sliding window on the logs.

Like link count, PageRank, anchor text, and other recommendation techniques, there is a potential bias in click data against new content. However, unlike simple query-independent click popularity counts, scores derived from click surrogates are query dependent and therefore generally capable of more rapidly responding to changes. Consider the hypothetical case of a highly controversial government report 'the Cierpinski Report', whose publication on a government website causes a massive increase in popularity of the query 'Cierpinski Report'. Provided that the baseline search algorithm was able to return the report at a point in the ranking visible to some searchers, click evidence linking the query with the document would build up. Using simple query-independent click frequency, it would take a very long time to compete with other popular documents. On the other hand, using query-dependent click data, the desired answer can potentially very quickly overtake other candidates. The temporal sliding window method helps to limit bias against new content in the case of queries with ongoing popularity.

Problems of link redirection and of links to documents eliminated by a duplicate detector are much less of a problem for click data than for link measures and anchor text. On the other hand, unlike anchor text, click evidence cannot provide descriptions of documents external to the crawl.

To achieve success for all types of query and for tasks other than best-document finding, we believe that click description scores should be combined with scores from anchor text and content and with query-independent measures. Xue et al. [24] report a simple fusion technique for combining with content while Craswell et al. [7] propose methods for combining query dependent and query independent evidence.

Click description surrogates depend upon an initial baseline ranking and are to some extent limited by its failings. We have found that click evidence is capable of promoting documents from deep in the original result rankings and also, due to query word overlap, to respond to queries never previously typed. It is not clear whether occasional random perturbations of rankings as practised by DirectHit would lead to better click descriptions.

Although we have demonstrated substantial improvements over the baseline (and over anchor text surrogates) for popular queries on a best-document finding task, the greatest potential gain from click data may lie in non-web environments, where link measures and anchor text are unavailable. Examples of such environments include library subject catalogues, and search of personal or corporate email and office documents.

6 Conclusion

We show that click-implied description surrogates alone can support good performance on best-document finding tasks in four very different webs. Using these surrogates, a mean reciprocal rank score of over 0.5 is achieved for popular queries in three out of four test corpora.

For the popular query sets, rankings based on click surrogates alone significantly outperform the original baseline ranking for three out of four corpora. They also outperform the ranking derived from Okapi BM25 scoring of anchor text surrogates.

We find that surprisingly little click data is necessary to achieve good results for popular queries and that performance on the best answer finding task approaches an asymptote once sufficient data is available. Click description surrogates are consequently very small, leading to efficient calculation of retrieval scores.

Comparison of query word and query token surrogates for the stock exchange sitemap set of queries shows a major benefit (92% relative gain in MRR) due to query word overlap. By contrast, there was no significant benefit on the popular query set, comprising much shorter queries with much more click evidence available.

Many interesting avenues await further research including: development of more sophisticated analytical models; methods for combining click surrogate scores with other ranking information; determining whether there is additional value in using click data as query independent evidence; spam rejection techniques; and investigating the use of clicks in non-web applications.

References

[1] Eugene Agichtein, Eric Brill and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proc. SIGIR*, 2006.

[2] Eugene Agichtein, Eric Brill, Susan Dumais and Robert Rango. Learning user interaction models for predicting web search result preferences. In *Proc. SIGIR*, 2006.

[3] Einat Amitay, Adam Darlow, David Konopnicki and Uri Weiss. Queries as anchors: selection by association. In *Proc. HYPERTEXT*, 2005.

[4] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In *Proc. WWW*, 1998.

[5] J. Carriere and R. Kazman. Webquery: Searching and visualizing the web through connectivity. In *Proc. WWW*, 1997.

[6] Nick Craswell, David Hawking and Stephen Robertson. Effective site finding using link anchor information. In *Proc. SIGIR*, 2001.

[7] Nick Craswell, Stephen Robertson, Hugo Zaragoza and Michael Taylor. Relevance weighting for query independent evidence. In *Proc. SIGIR*, 2005.

[8] Gary Culliss. User popularity ranked search engines, 1999. `http://web.archive.org/web/20000302121422/http://www.infonortics.com/searchengines/boston1999/culliss/index.htm`.

[9] Pavel A. Dmitriev, Nadav Eiron, Marcus Fontoura and Eugene Shekita. Using annotations in enterprise search. In *WWW*, 2006.

[10] David Hawking, Nick Craswell, Francis Crimmins and Trystan Upstill. How valuable is external link evidence when searching enterprise webs? In *Proc. ADC*, 2004.

[11] David Hawking, Trystan Upstill and Nick Craswell. Towards better weighting of anchors (poster). In *Proc. SIGIR*, 2004.

[12] David Hawking and Justin Zobel. Does topic metadata help with web search? *JASIST*, 2006. (To appear.).

[13] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proc. ACM KDD*, 2002.

[14] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. ACM SIGIR '05*, 2005.

[15] Ming Lei, Jianyong Wang, Baojue Chen and Xiaoming Li. Improved relevance ranking in webgather. *Journal of Computer Science and Technology*, Volume 16, Number 5, 2001.

[16] Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998. `http://dbpubs.stanford.edu:8090/cgi-bin/makehtml.cgi?document=1999/66`.

[17] S. E. Robertson, S. Walker, M.M. Hancock-Beaulieu and M. Gatford. Okapi at TREC-3. In *Proc. TREC-3*, 1994. NIST spec. pub. 500-225.

[18] Tetsuya Sakai and Karen Sparck-Jones. Generic summaries for indexing in information retrieval. In *Proc. SIGIR*, 2001.

[19] Falk Scholer, Hugh E. Williams and Andrew Turpin. Query association surrogates for web search. *JASIST*, Volume 55, Number 7, 2004.

[20] Barry Smyth, Evelyn Balfe, Jill Freyne, Peter Briggs, Maurice Coyle and Oisin Boydell. Exploiting query repetition and regularity in an adaptive community-based web search engine. *User Modeling and User-Adapted Interaction*, Volume 14, Number 5, 2005.

[21] Danny Sullivan. Search engine sizes, 2005. [Online; accessed 24 Jan 2006] `http://searchenginewatch.com/reports/article.php/2156481`.

[22] Wikipedia. Google bomb—Wikipedia, The Free Encyclopedia, 2006. [Online; accessed 21 Jan 2006; `http://en.wikipedia.org/w/index.php?title=Google_bomb&oldid=36057937`].

[23] Gui-Rong Xue, Shen Huang, Yong Yu, Hua-Jun Zeng, Zheng Chen and Wei-Ying Ma. Optimizing web search using spreading activation on the clickthrough data. In *Proc. WISE*, Volume LNCS 3306, 2004.

[24] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi and WeiGuo Fan. Optimizing web search using web click-through data. In *Proc. CIKM*, 2004.

My Instant Expert

George Ferizis
CSIRO ICT Centre
Locked Bag 17
North Ryde NSW 2113 AUSTRALIA
george.ferizis@csiro.au

Peter Bailey
CSIRO ICT Centre
GPO Box 664
Canberra ACT 2601 AUSTRALIA
peter.bailey@csiro.au

Abstract *This paper gives an overview of a mobile device question answering application that has recently been developed in the CSIRO ICT Centre. The application makes use of data in an open-domain encyclopaedia to answer general knowledge questions. The paper presents the techniques used, results and error analysis on the project.*

Keywords Information retrieval, Question answering

1 Introduction

My Instant Expert™is a mobile device question answering system. The aim of the application is to create a general knowledge question answering system that can be accessed from a mobile device. It uses Wikipedia [2] data to answer questions.

Many question answering systems already exist [6, 8]. *My Instant Expert*™is different to most other question answering systems as it answers open-domain questions from a closed corpus [1], it has tight time constraints on pre-processing and query processing, and it delivers to mobile devices.

Answering open-domain questions from a closed corpus presents many challenges, as it is difficult to construct open-domain ontologies or taxonomies that would aid question answering. The small amount of text redundancy in the corpus makes it challenging to retrieve the correct text segments while searching for answers. Applying computationally expensive language processing techniques, such as coreference resolution, which could potentiall improve the accuracy of the system, cannot be applied due to the tight time constraints on pre-processing and query processing stages.

This paper begins by discussing the development methodology behind the project before giving an overview of the components of the system. The results of system testing using questions from previous TREC question answering tracks [16] is then presented, along with discussion on these results and some error analysis.

The paper continues with a discussion of some of the challenges related to developing mobile device applications, and presents an overview of the mobile device application that we developed. Particular focus is given to how the application differs from an equivalent application deployed via a web browser. A description of a user evaluation which was conducted using the mobile phone application is then given, and the results of the evaluation are discussed. The evaluation was conducted to understand user preferences for question answering on mobile devices. The final section discusses some future research directions that result from the error analysis and the user evaluations.

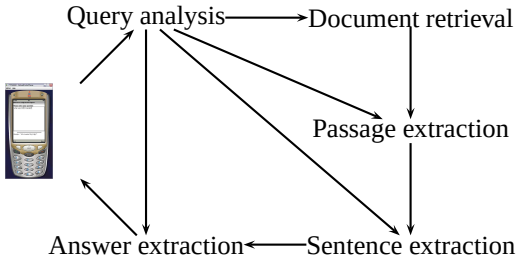


Figure 1: System diagram

2 Description of work

My Instant Expert™does what a number of question answering systems do. It sequentially applies more computationally expensive search techniques to reduce the search space from the entire corpus to a small number of candidate answers. Figure 1 contains a diagram of the system. In order of application these techniques are: document retrieval, sub-document passage retrieval, sentence extraction, and answer extraction. An input to each of these stages is the output of the previous stage and the results of some query analysis. Communication is done through the transfer of XML files between modules. This allows modules to be substituted with others easily.

The original system design envisioned an application that would run on a mobile platform. Unfortunately as the size of the Wikipedia corpus is currently doubling annually and contains 6GB of data as of September

[1] The MIT START system is open domain, but it is restricted to a set of queries; see *When did John Howard become prime minister?*

Proceedings of the 10th Australasian Document Computing Symposium, Sydney, Australia, December 12, 2005.

2006, device constraints make this task very difficult. As a result, we developed a client server architecture where the mobile device application queries a server which then processes the natural language questions.

2.1 Query analysis

The query analysis module identifies what "topics" or phrases the query is asks about. This is done by chunking the text using the OpenNLP toolkit before removing chunks that contained words to do with question structure (e.g., "*what is*" and "*when did*").

The type of answer required (*expected answer class*) is also determined. The set of answer classes is restricted to: *location*, *numeric*, *person*, *date* and *description*. Everything else is placed into an *unidentified* class. The questions are classified using a Bayesian classifier over features such as named entities in the query, term frequency of words from specific dictionaries, part of speech bigrams and chunking bigrams.

Query expansion is also done during this stage. During query expansion, terms that are semantically related to terms in the query are discovered using WordNet [3]. The kinds of relationships found are synonym, hyponym, derivational and attribution relationships.

2.2 Document retrieval

Typically question answering systems use a document retrieval engine to retrieve documents from the corpus that may contain a possible answer. Experimentation of various ranking techniques using the wikipedia corpus found that retrieval on title metadata was superior to full-text retrieval for question answering. This lead us to test a naive method of document retrieval that ranks according to the presence of the document title in the question, with the score being the percentage of terms from the query occurring in the title. An example of this is shown in Figure 2 for the query "*When did John Doe eat cake?*" [2].

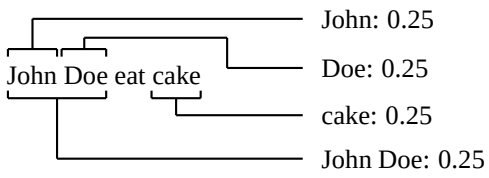


Figure 2: Ranking of documents against the query "*When did John Doe eat cake?*"

A comparison of this technique with several other techniques is presented in Table 1. Google queries were of the form "`site:en.wikipedia.org (intitle:<CHUNK>)+`", when querying on title metadata and "`site:en.wikipedia.org (<CHUNK>)+`", when querying on full text. The queries to the other search engine were of similar form. The table shows that the method of "*title matching*" outperforms both full text search, and probabilistic search over title metadata using two different search engines.

The reasons for this performance difference can be attributed to the task and the corpus. Wikipedia document titles are always unique and are more often than not named entities, or useful noun phrases that describe the contents of the document. Wikipedia documents also tend to have multiple titles, with over a third containing more than one title and over 3% containing over 5 titles.

2.3 Sub-document passage retrieval

Retrieved documents are broken into sub-document elements such as: image descriptions, lists, paragraphs, tables and templates (a form of MediaWiki marked up data). The elements are combined with contextual information about the position of the element in the document such as the section headings and document title.

Including the title and section headings allows us to approximate using a pronoun resolution tool. Wikipedia documents, like most documents, typically do not reference nouns repeatedly but instead use pronouns. Intuitively the noun that an encyclopaedia document will most frequently refer to using pronouns is the entity the document corresponds to. As such including the title and section headers when ranking a candidate sentence or passage (although weighted differently to terms occurring in the actual sentence) allows the contextual information of surrounding tags in the document to be included in the ranking of the sub-document elements.

The probabilistic Okapi BM25 [13] is used to rank passages from the retrieved document set. Stemming using the Porter stemming [12] algorithm was found to help the performance of the system.

2.4 Sentence extraction

The sentence extraction algorithm ranks sentences by comparing terms in the query and the candidate sentences. If the terms from the original question cannot be found in candidate sentences, the system searches for terms obtained from query expansion. Matches for terms obtained from query expansion are weighted lower during ranking than terms from the original sentence.

Tabular elements are parsed to disambiguate the contents of entities in the table. This is done using column and row headers if they are present in the table.

2.5 Answer extraction

If the expected answer class can be identified, then the answer extraction module attempts to find the matching named entity in the candidate sentences. This both reranks the sentences if the named entity is detected and extracts the desired named entity from the sentence. The module does this through named entity recognition,

[2] Due to space constraints a real query and real Wikipedia documents are not used in the example.

Table 1: Comparison of title matching with various other methods

Method	P@1	P@5	MRR
Title matching	0.77	0.85	0.81
Full-text search	0.44	0.62	0.53
Full-text search - Nouns only	0.42	0.62	0.51
Full-text search using title metadata	0.31	0.48	0.38
Full-text search using title metadata - Nouns only	0.33	0.58	0.43
Google	0.25	0.35	0.29
Google - Nouns only	0.48	0.56	0.52
Google using title metadata	0.42	0.56	0.49
Google using title metadata - Nouns only	0.42	0.58	0.50

using a tool base on the GATE toolkit [7] and some shallow sentence parsing to identify similar syntactic relationships between terms in the query and terms in the sentence.

The results returned contain a short answer, which corresponds to the named entity regarded to be the answer, and a longer answer to justify the shorter answer (typically the sentence in which the shorter answer was found). These two separate answers are used for display purposes on smaller mobile device screens. If the named entity is not found in the sentence, or if the named entity required by the question is unidentified, the answer only contains the short answer which corresponds to the candidate sentence. Figure 3 shows sample output in response to two questions.

```
<original>
    When was John Howard born?
</original>
<short-answer>26 July 1939</short-answer>
<detailed-answer>
    John Winston Howard (born 26 July
    1939) is an Australian politician and
    is currently the Prime Minister of
    Australia.
</detailed-answer>

<original>What color is grass?</original>
<short-answer>
    Grass generally describes a
    monocotyledonous green plant in the
    family Poaceae, botanically regarded
    as true grasses.
</short-answer>
<detailed-answer />
```

Figure 3: Sample output

3 Special circumstances

There are two circumstances where *My Instant Expert*™does not follow the steps outlined above to answer a question. The first involves answering definitional questions such as "*Who is X?*" and "*What is Y?*". In the absence of any contextual information that could be used to infer points of interest about *X* or *Y*, the answer returned is a summary of the Wikipedia document describing the entity. As Wikipedia documents typically start with a brief summary of the document, the returned answer is usually the first paragraph of the document.

The second involves disambiguation of entities. Again, in the absence of any contextual information, it is difficult to answer question such as "*Who is George Bush?*", or "*What is a plane?*", as the names used in either question can refer to multiple entities. To solve this, a list is presented to the user containing entities that match their query. The user then obtains an answer by selecting the entity they are interested in.

4 Results

Automated result quality testing using the TREC QA track test sets [16] was conducted during development. As the TREC QA test set contains many questions that are temporally constrained (e.g., *What is the population of Mississippi?*), a snapshot of the Wikipedia corpus was taken, and all answers were verified using the corpus.

During testing, over 800 questions were used. We took measurements of precision at 1 (*P*@1) - how often was the first answer a right answer, and the mean reciprocal rank (MRR) - on average what was the rank of the highest ranked correct answer. A response from the system was considered correct if it contained one of the answers that was manually identified as correct. As no attempt was made to construct sentences for answer responses, an answer returned by the system was considered correct if it contained all of the terms in the correct answer. The overall system performance found that the correct answer was ranked first 36% of the time, and that the MRR was 0.50.

The graph in Figure 4 displays the *P*@1 and *MRR* results for the document retrieval, passage extraction, sentence extraction and answer extraction phases of the system. As is expected, the accuracy of the system decreases as the amount of data being returned by the system decreases, with retrieval of large quantities of

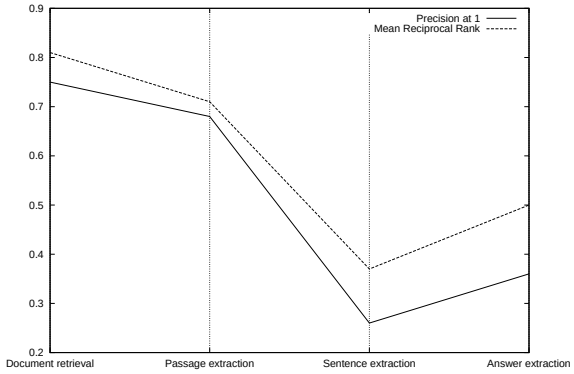


Figure 4: Results of testing by phase

text such as documents and passages having a high accuracy. The reranking of answers based on the presence of the requisite answer class in the final stage improves the accuracy of the system significantly.

The accuracy gains of the answer extraction phase supports the assertion that named entity recognition impacts on the performance on the system. Table 2 shows that, with the exception of the *mountain* entity (which can be attributed to a statistical anomaly due to infrequent occurence), the answer tended to be more accurate when the entity required by the question could be easily discovered by the named entity recogniser.

Table 2: Results of testing, grouped by answer class

Answer class	**P@1**	**MRR**
Group	0.67	0.70
Mountain	0.50	0.55
Description	0.49	0.67
Date	0.47	0.63
Individual	0.44	0.55
City	0.38	0.49
Country	0.33	0.39
Unidentified	0.32	0.45
Numeric	0.12	0.29
Title	0.07	0.17
State	0.00	0.23
Overall	0.36	0.50

Our results compare favourably with recent results from the TREC QA track, ranking in the top 10 performers on *initial factoid* questions, although it is unlikely that our assessment was as strict as that used in the TREC track. There were differences to how our system and the systems for the TREC track were evaluated:

- TREC participants did not need to implement document retrieval - they were given a list of documents from the corpus that could possibly answer the question.
- The TREC corpus is fixed, however precedent to make use of external corpora such as Wikipedia has been established [4].
- It is not clear how precise an answer must be before it is considered correct for the TREC track, e.g., is *Mississippi* a suitable substitute for the answer *Mississippi river*?

5 Discussion

The results showed that the system may be improved if the accuracy of the named entity recogniser was higher. Further error analysis showed how vital the accuracy of the question classifier is.

Table 3: Results of testing, grouped by correct identification of answer entity

Identification	Number	**P@1**	**MRR**
Correct	307	0.45	0.59
Incorrect	504	0.31	0.46
Overall	811	0.36	0.50

The results in table 3 demonstrate that when the required answer entity was correctly identified the performance of the system increased dramatically. The results also highlight how inaccurate our question classifier was.

Improvements to the answer classifier must address two abstract questions: "*How many types of entities should be detected?*" and "*How fine grained should these entities be?*".

5.1 Question classification vs. named entity recognition

Named entity recognisers and classifiers for closed domain applications, such as the medical or military domains, focus on classifying entities into a large set of very specific manually developed categories which are derived from a small and closed set of broad categories. As *My Instant Expert*^{TM}is an open domain application, questions could be about anything. Thus it is not feasible to develop manually an exhaustive set of non-overlapping entities that make sense. Even if this set was developed, the task of creating a classifier would be more difficult as having more categories and fewer distinctions between the categories makes classification much less accurate.

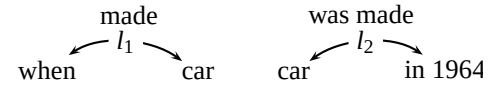
Research into classification of questions [9] from the TREC QA track [16] shows that results of up to 84% can be obtained for the finest classification, over a variety of entities ranging from vehicles to currency. Inspection of their methods is possible as they have generously provided the training data and pre-processing information on-line [1]. It shows that the feature selection is very specific to their categories, e.g., they have specific rules for colours. This would suggest that if more category specific features were included more accuracy gains could be made. However, with a large set of categories, how would these features be obtained?

Being able to classify the question is only halfway to solving the problem. Even if a question classifier

could always determine what entity a question is asking for, the named entity recogniser and classifier still needs to find the entity in the candidate answers. The task of identifying the entities becomes even more difficult. When there are a larger number of entities. The errors of the two stages compound each other, when using current state of the art techniques and a small set of entities, the error rate is such that 1 in 5 questions will have an error as a result of either of these two processes. Without improvements to either the question classification or named entity recognition this error rate will remain quite high.

5.2 Semantic relationships

Techniques that attempt to discover syntactic and semantic relationships between words [14] may reduce the dependency of the system on the problem of entity classification. Consider the example in Figure 5. If the relationships in Figure 5(a) and Figure 5(b) could be discovered and if similarity between the labels for the relationships l_1 and l_2 could be inferred, there would be no need for the system to recognise that the question is asking for a date.



(a) *When was the car made?* (b) *The car was made in 1964.*

Figure 5: Example relationships

The example given is a basic one, and the graphs in the relationships unrealistic. However it is illustrative. It is trivial to infer that the term `made` is identical to the term `made`. However, how easy is it to infer that the term `manufactured` may have similar meaning? How much harder still is it to infer that the phrase `rolled off of the assembly line` may also be equivalent in some circumstances? The problem of discovering semantic relationships such as these contributes significantly to the error rate of the system. While it is hard to put a figure on this contribution, manual experimentation suggests that it could be quite high.

It is hard to quantify the effects of missing semantic relationships. Our results show that the right answer was ranked at 1, 68% of the time during the ranking of paragraphs, while this result drops to 26% when the unit of text is sentences. A substantial part of this decrease in performance is likely due to the text redundancy present in the larger passages, hiding the inability of the system to discover semantic relationships between terms in the query and terms in the passage, such as anaphora. The importance of resolving anaphora has already been measured in previous work [15].

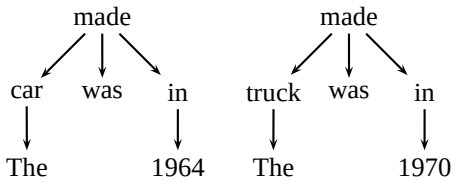
Previous measurement into the negative contribution of undetected synonymy on question answering performance [6] suggests that it contributes to only 2% of the incorrectly answered questions. However, these measurements used the web as the corpus. Using the web as a corpus makes the case for detecting synonymy far less compelling as, due to the size of the web, it is likely that the same relationship has been expressed using many different phrases. Manual experimentation suggests that the inability of the system to detect other semantic relationships such as synonymic, metaphoric and metonymic[3] relationships between query terms and phrases in potential answers results in a large number of poorly answered questions.

WordNet is used in *My Instant Expert*™ to detect some of these relationships, and it works very well for detecting term-to-term synonymy and even well for some term-to-phrase synonyms such as `kicked the bucket` and `die`. However, it is far from complete. If, for example, the question *"When was the first Ford Mustang manufactured?"* was submitted, we are incorrectly informed that it was manufactured in 1972. Examining the Wikipedia corpus, we could find an answer to this question only in the phrase *"The first production Mustang, .., rolled off the assembly line in Dearborn, Michigan on March 9, 1964"*. If the metonymic relationship between `manufactured` and `rolled off the assembly line` was detected, this would have been a highly ranked answer. But unfortunately, it was not.

In 2001, Lin and Pantel [10] described a method that measures syntactic relationships between two phrases to determine whether two phrases are similar. This method could be used to populate databases like WordNet automatically. The method works by identifying relationships between parse trees such as those shown in Figure 6. Using a starting phrase of "`The car was made in 1964`", the algorithm will determine that the paths between {`car, 1964`} and {`truck, 1970`} are similar as they follow a similar route through their parse trees (`made in`).

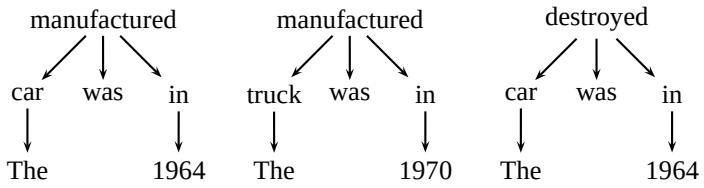
The next step of the algorithm involves it trying to discover all other statistically significant paths that connect the terms {`car, 1964`}, and {`truck, 1970`}. Figure 7 shows some phrases that will likely be matched by this phase of the algorithm. The three trees in Figures 7(a), 7(b) and 7(c) all would be returned. While on first appearance this is incorrect, it is in fact correct as `destroyed` is in some way related to the term `made`. Without a significantly higher rate of occurrence of the other two trees, it is very likely that `destroyed` will be related to `manufactured`. While this is correct it is not the desired outcome. The initial work attempts to leverage textual redundancy which does not exist in Wikipedia. Preliminary results of using this technique over the Wikipedia corpus has found that there is not enough text redundancy to filter out antonym-like semantic relationships. The results

[3] metonymy: A figure of speech in which one word or phrase is substituted for another with which it is closely associated, as in the use of `the crown` for the English monarchy, or `Washington` for the United States government.



(a) Parse tree for *The car was made in 1964.* (b) Parse tree for *The truck was made in 1970.*

Figure 6: Similar parse trees



(a) Parse tree for *The car was manufactured in 1964.* (b) Parse tree for *The truck was manufactured in 1970.* (c) Parse tree for *The car was destroyed in 1964.*

Figure 7: Possibly similar parse trees

however has only processed a 10% of Wikipedia due to time and space constraints.

6 Mobile application

Developing applications for mobile devices presents many problems that are not encountered when developing browser based applications. Different manufacturers of mobile devices create vastly different APIs for their devices, and devices from the same manufacturer often have significantly different APIs. While this problem can be solved by using portable APIs such as Sun's J2ME, to remain portable the APIs do not take full advantage of the device's I/O capabilities. Even basic capabilities such as horizontal screen scrolling cannot be taken for granted or joystick functionality. As a consequence, developing an application that remains both portable and usable is especially difficult.

Our application was developed using J2ME and is compatible with any mobile devices that are MIDP 2.0 and CLDC 1.0 compliant. We attempted to get around constraints on the use of phone I/O capabilities by making novel use of rudimentary API functions.

Our application allows the user to enter a new question or rephrase an existing question using the default SMS-style input interface of the mobile device as shown in Figure 8(a). This allows the input aspect of the application to remain as portable and as familiar to the user as possible.

While it is possible to present a large list of possible answers that are very verbose in a web browser, due to the space constraints on the screen of a mobile device, this is not possible. This forced us to reconsider two common design aspects of question answering interfaces and search interfaces: how the justification of an answer is presented, and how multiple answers or results are presented to the user.

As shown in Figure 8(b) the answer to the question "*When was CSIRO founded*" is split into two portions, one containing a short answer (1926), and the other containing the text this answer is obtained from. This text is included to justify the answer to the user. Splitting the text like this gives the user the ability to read an answer without having to scroll through multiple screen folds if they do not desire justification.

The screen shots in Figures 8(b) and 8(c) show different answers to the same question being displayed on the mobile device screen. These answers are selected through the use of a tab control on the top of the screen. This differs from the usual result list presented by most search engines. This allows users to quickly access different answers without a need to scroll through many screen folds.

7 Usability and system evaluation

A central concern when designing any application is usability. We developed a series of user stories and personas to help identify the kinds of environments in which the application would be used. These provided input into the initial user interface design prototype for the mobile device.

(a) Question input screen. (b) First answer screen. (c) Fifth answer screen.

Figure 8: Screenshots of the mobile device client

Various usability evaluations were carried out through the life of the project. These included early think-aloud evaluations of the prototype mobile device UI, which identified various problems with the initial design. Once the system was operational, we discovered technical problems with answer quality and mobile device UI software development kits that required additional changes. We followed these later with more comprehensive whole-system evaluations with a new set of individuals. A survey questionnaire was developed for the latter to help identify familiarity of the evaluator with mobile device technology in general. Again, a number of issues were picked up, leading to improvements in the design of the interaction processes and layout.

System evaluation was conducted on the users after the usability evaluation. The users were asked a series of questions revolving around their "feelings" about the system. The questions included:

- Did they feel the system met their information need?
- Was the information returned in a timely manner?
- How highly did they value justification of the answer?
- Did they enjoy reading related information?
- Overall satisfaction with the system.

Several results of the evaluation were surprising:

- While users preferred the short answer being displayed on the screen first, 90% of users appreciated having a longer answer present on the mobile device even if it meant having to scroll repeatedly.
- Often enough only 1 out of 5 of the answers was correct, the corollary of this being that the remaining 4 were incorrect. Surprisingly so long as the answers were on topic the users liked it. The reasoning for this was that the system had provided them with more and still relevant information which made them feel satisfied that they had obtained useful information from the system.
- Users did not mind waiting up to 2 minutes for an answer to be returned, even though the communication was not asynchronous.

The first two results suggest that presenting serendipitous information, or giving easy ways to to access such information is judged to be favourable by users. The third results suggests that applying more computationally expensive techniques to answering questions may be feasible without raising the ire of users. Overall the system was liked by users.

8 Scope and plans for future research

The *My Instant Expert*^{TM}architecture also supports ongoing enhancements to components and functionality extensions, which serve the goal of building a platform for research experiments. Again due to the short development timeframe, some of the research experiments involving contextual delivery of information could not be carried out. For example, do alternative methods of presenting information make a qualitative difference to users and. can prior conversational history be used to answer follow-up questions better? These can be investigated subsequently using the basic system.

The dilemma faced by researchers interested in QA systems is that nearly all existing QA test collections are not reusable. (An exception is the factoid question test collection developed by Lin and Katz [11].) The reason for this is that judged documents (and ensuing relevance measures) from a corpus are rarely sufficient to support stability of the measures when new systems discover additional documents (not previously judged).

As mentioned in [5], we plan to provide logs (suitably anonymised) of real user queries from the system to the broader research community. These can be used to understand what real users ask of an open domain QA system. From a research perspective, access to real user query logs is an invaluable resource, and one that is often available only to the operators of search services. These query logs will provide one component required for the development of additional reusable QA test collections.

The query logs can also be used to work towards a solution for one of the problems identified during error analysis. Inspection of query logs may allow us to develop a taxonomy of entity classes specific to open-domain question classification based on the kinds of queries users frequently make.

As raised during the system evaluation, users valued extra on-topic information. Research into how to determine what would be appropriate "follow-up" questions to present the user with is currently being conducted. The research is looking at what questions would be valuable to users based on the question asked and the answer received.

9 Acknowledgements

We would like to acknowledge various members of the CSIRO ICT Centre for their contribution to this project: Alexander Krumpholz, Ronnie Ma, Denis Mikhalkin, Shannon O'Brien, Cécile Paris and Tom Rowlands. We also appreciate the input on other matters relating to the project from Jim Lilley, Ross Wilkinson, David Hawking, Colin Murphy, David Lau, Anthea Roberts, Andrew Lampert, Pascale de Souza Dromund, Tom McGinness, and Gautam Tendulkar. Special thanks to Alex Zelinsky for sponsoring the project in the first place.

References

[1] Learning Question Classifiers - Experimental Data for Question Classification. *http://l2r.cs.uiuc.edu/ cogcomp/Data/QA/QC/*.

[2] Wikipedia. *http://www.wikipedia.org*.

[3] *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.

[4] David Ahn, Valentin Jijkoun, Gilad Mishne, Karin Muller, Maarten de Rijke and Stefan Schloback. Using Wikipedia at the TREC QA track. In *TREC*, 2004.

[5] Peter Bailey and George Ferizis. Possible approaches to evaluating adaptive question answering systems for mobile environments. In *First International Workshop on Adaptive Information Retrieval (AIR)*, Glasgow, UK, October 2006.

[6] Eric Brill, Susan Dumais and Michele Banko. An analysis of the askmsr question-answering system. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 257–264, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[7] H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[8] Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy J. Lin, Gregory Marton, Alton Jerome McFarland and Baris Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *NLDB '02: Proceedings of the 6th International Conference on Applications of Natural Language to Information Systems-Revised Papers*, pages 230–234, London, UK, 2002. Springer-Verlag.

[9] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[10] Dekang Lin and Patrick Pantel. Discovery of inference rules for question-answering. *Natural Language Engineering*, Volume 7, Number 4, pages 343–360, 2001.

[11] Jimmy Lin and Boris Katz. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology*, Volume 57, Number 7, pages 851–861, 2006.

[12] Martin Porter. An algorithm for suffix stripping. *Program*, Volume 14, pages 130–137, 1980.

[13] S.E. Robertson, S. Walker and M. Beaulieu. Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track.

[14] Dan Shen, Geert-Jan M. Kruij and Dietrich Klakow. Exploring syntactic relation patterns for question answering, 2005.

[15] José L. Vicedo and Antonio Ferrández. Importance of pronominal anaphora resolution in question answering systems. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 555–562, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[16] Ellen M. Voorhees. Question answering in TREC. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 535–537, New York, NY, USA, 2001. ACM Press.

Preliminary Investigations into Ontology-based Collection Selection

J. D. King, Y.Li, P.D. Bruza and R.Nayak
School of Software Engineering and Data Communications
Queensland University of Technology
QLD 4001 Australia
{*j5.king, y2.li, p.bruza, r.nayak*}*@qut.edu.au*

Abstract *This article tackles the collection selection problem from the query side. Queries are enhanced by mapping them to subjects in an ontology; the associated subject classification terms are then employed to retrieve collections. An experimental comparison was performed with the state of the art ReDDE system, which relies on estimates of collection size to rank collections. Although the research is preliminary, there is some support to the hypothesis that this approach mitigates the need for collection size estimates in collection selection.*

Keywords Information Retrieval, Document Databases, Digital Libraries

1 Introduction

Currently human experts are better at identifying relevant documents than the state of the art information retrieval methods. Human experts are also currently better at classifying documents than the state of the art automatic classification methods. One factor that makes human experts superior from computer programs is '*world knowledge*'. World knowledge encompasses information on topics such as philosophy, psychology, religion, social sciences, language, natural sciences, mathematics, technology, the arts, literature, geography, and history. In this study we make use of world knowledge stored in an ontology and apply it collection selection. The term "ontology" has a number of conceptions. For the purposes of this article, an ontology is defined to be a hierarchical structure, whereby the nodes correspond to subjects. Each subject is characterized by a set of subject classification terms.

Ontologies have been used in Artificial Intelligence for a variety of applications. However, a major problem associated with building an ontology which covers a large number of domains is the human-hours that would be required to construct it. This problem is called the *knowledge acquisition bottleneck*. The aim of this research was to quickly, cheaply and simply build an ontology which has both a wide range of knowledge and capabilities across many different domains. While some information retrieval systems use terms to describe collections, our method uses subjects to describe collections. The power of a subject based approach is better understood through the following example. If a user issues the query "matrix factorisation methods" into a search engine, he or she would probably expect documents about "singular value decomposition" to be returned. In this article, we attempt to exploit this capability in the following way: An arbitrary query is mapped into the ontology yielding a set of subjects. The subject classification terms of each subject are then accumulated into a query which is used to rank collections. It is important to note that this approach does not rely on estimating the collection size. Even though collection size is acknowledged as being an important feature determining collection selection effectiveness [31, 30], it is also acknowledged that acquiring reliable estimates can be a costly and challenging problem. The hypothesis behind this article is to examine whether a subject based approach may compensate for not having collection size estimates. In other words, we are tackling the collection selection problem for the query side, rather than the collection side.

The rest of this paper is organised as follows. Section 2 introduces our automatic ontology learning method, Section 3 shows our collection selection method, Section 4 shows related work, Section 5 shows our experiment data, Section 6 gives our experiment results, and Section 7 concludes the paper.

2 Automatic Ontology Learning

The problem with many ontologies is that they only cover a small number of domains, and each domain has to be manually added by a domain expert. The method presented in this section automatically creates an ontology covering hundreds of different domains. Automatic ontology learning will be a great improvement, enabling technologies to facilitate the creation of the semantic web.

There are three methods of ontology learning, each offering a trade-off between speed and accuracy. The three methods are:

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.

1. to generate rules from free text (fast but inaccurate)
2. to generate rules from expert created and/or classified materials such as dictionaries and encyclopedia texts
3. ask domain experts to populate the ontology by manually entering rules (slow but arguably accurate)

The second method is adopted in this research as it provides a balanced approach.

The stages involved in the ontology construction process are:

1. Selecting a classification taxonomy
2. Identifying a training set
3. Downloading a training set and populating the ontology
4. Cleaning up the ontology

We refer to this as the "IntelliOnto" construction process. (See [17] for more details).

Ideally there are several desirable properties in a good expert classified taxonomy. The taxonomy should cover a wide number of subjects, be carefully constructed, be standard across the world, and be available in different languages. It was decided to use the Dewey Decimal System, a widely used library classification system[1]. The system has been used for the classification of a large number and wide range of materials. The Dewey taxonomy is based on a pattern of ten root nodes, with each root node having ten child nodes. Each child node has another ten child nodes with this pattern continuing downwards. There can be many different levels of the taxonomy, depending on how precise the subject match is. There are 1,110 classification nodes at the top three levels of the taxonomy, with many more nodes in the lower levels of the taxonomy. There are some low-level subject nodes that are unused because of depreciation or limited coverage. In this paper only the top three levels of the taxonomy are used.

Figure 1 shows part of the Dewey taxonomy, and Figure 2 shows a more detailed portion of the taxonomy. Each Dewey Decimal Code (DDC) provides the best possible classification for each item.

The desirable properties of a training set are that it is large, of high quality, and covers a wide range of subjects. A data set reflecting these requirements is the Queensland University of Technology Library Catalogue[2], which contains over 500,000 usable items, although we only sampled 80,000 items for this research.

[1]For a full listing of the classifications see http://www.tnrdlib.bc.ca/dewey.html.

[2]See http://libcat.qut.edu.au/ This library web site is excellent for use as a training set because most of the entries have extra meta-information such as descriptions and summaries.

term	term count
software	281
programming	205
security	200
program	191
web	152
object	117
database	117
programs	105

Table 1: Terms that occur most frequently in *005 Computer programming, programs, data*

(It should be noted the Queensland University of Technology Library Catalogue is but an exemplar of an ontology which can be employed. The IntelliOnto method is by no means tied to this particular source of knowledge).

This data set was used to populate the ontology with world knowledge. Each document in our training set is assigned a Call Number. These documents have been carefully classified by experts in the field, and the information is of superior quality to other web based directories.

2.1 Mining From the IntelliOnto Ontology

Once the ontology base has been constructed, classification rules are mined from it. These rules are then used to classify collections. There are many different classification rules that can be mined from the ontology by using the terms, the subjects, and the taxonomic structure. By finding patterns between subject nodes and terms we are able to extract classification rules. These rules can then be made more useful by applying the taxonomic nature of the Dewey Decimal system.

The subject classification terms characterizing a subject need to be carefully selected. These terms should preferably be subject-specific (occurring within few or no other subjects) and should occur frequently within the subject and infrequently in other subjects. It is difficult to decide which terms to select as there are many possible terms to describe a subject. Many terms may not occur in common English dictionaries yet are still valuable for classification. These may include technical or subject specific terms such as conference names, acronyms and names of specialist technology. Some examples from computing are *RMI*[3], *SMIL*[4], *XSLT*[5], and *servlet*[6]. Few standard English dictionaries include these terms, yet if any of these acronyms occur in a document it is likely the document covers a subject related to computing.

Our first term selection method, highest term frequency, involves selecting the most popular terms from

[3]Remote Method Invocation.

[4]Synchronized Multimedia Integration Language

[5]Extensible Stylesheet Language Transformation.

[6]"*A Java application that, different from applets, runs on the server and generates HTML-pages that are sent to the client*" http://www.softwareag.com/xml/about/glossary.htm

Term	Count	Support	Confidence
c#	55	0.00003840	1
j2ee	48	0.00003351	1
javabeans	43	0.00003002	1
fedora	27	0.00001885	1
sax	27	0.00001885	1
awt	25	0.00001745	1
xsl	23	0.00001606	1
jdbc	23	0.00001606	1
oo	20	0.00001396	1
unicode	20	0.00001396	1

Table 2: Terms for *005 Computer programming, programs, data* with a confidence score of one.

each subject. Table 1 shows the most frequent terms for the subject *005 Computer programming, programs, data*. Our second term selection method, highest support and confidence, involves finding the most distinguishing (or unique) terms from each subject based on confidence and support. Table 2 shows the most distinguishing terms for the same subject. These terms cluster around the Dewey Decimal code "005". The nodes are grouped based on the third level of the taxonomy, any groupings below this level are not considered.

A term-subject pair $p(t \rightarrow s)$ in $M(O)$ with their confidence and support values is referred to as a pattern $p(t \rightarrow s) :=< t, s, conf(t \rightarrow s), sup(t \rightarrow s) >$ in this paper, where $t \in T, s \in S, conf(t \rightarrow s) = [0, 1]$ and $sup(t \rightarrow s) = [0, 1]$. We use a modified support and confidence method for our system, in order to accommodate the taxonomy. The $conf(t \rightarrow s)$ and the $sup(t \rightarrow s)$ in the pattern describe the extent to which the pattern is discussed in the training set. The $conf(t \rightarrow s)$ and $sup(t \rightarrow s)$ are defined as follows:

$$conf(t \rightarrow s) = \frac{sf(t, s)}{sf(t)} \quad (1)$$

$$sup(t \rightarrow s) = \frac{sf(t)}{n} \quad (2)$$

where $sf(t, s)$ is the number of child subjects under s (including s) with t occurred in the $termset$. The greater $sup(t \rightarrow s)$ and $conf(t \rightarrow s)$ are, the more important the term t is to the subject s.

Of the two ranking methods, the terms selected with high confidence and support thresholds seemed to be better for collection selection than the terms selected by highest frequency. Some of the more frequent terms were so common across different subjects that they could virtually be considered stopwords. The results presented in this paper therefore only use the highest confidence and support method.

3 Collection Selection

Collection selection is the selection of an optimal subset of collections from a large set of collections for reducing search costs [4, 11, 15, 25, 7, 8, 14, 29, 13, 3]. A central aim of collection selection is to accurately classify the content of each collection being evaluated. Once the content of each collection has been determined, the best subset of collection can be returned to serve an information need[7].

By way of illustration take collections, called *Collection A* and *Collection B*. Collection A contains information on the *creative arts* and no information on *social science*. Collection B contains information on *social science* and less information on the *creative arts* than Collection A. Each collection is treated as a "black box" and no prior knowledge of the contents is assumed. A human expert is used to generate a set of significant

[7]Many collection selection methods require direct access to or communication with each collection, yet few internet collection allow this. Thus other methods of evaluating collection content must be developed.

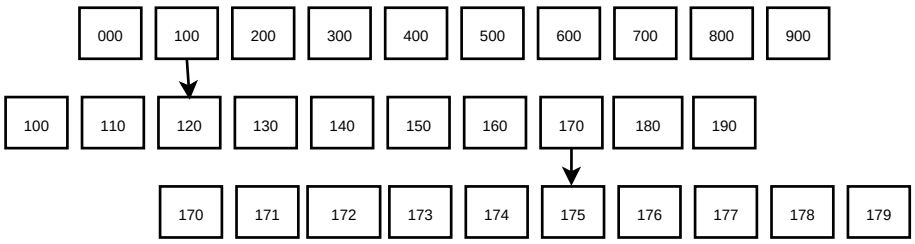


Figure 1: The Dewey Decimal taxonomy

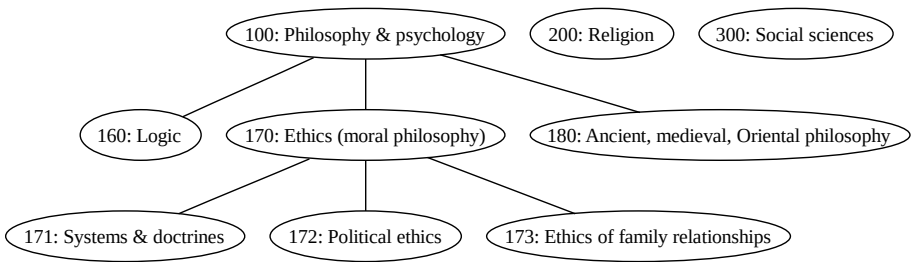


Figure 2: A Portion of the taxonomy

classification terms for each subject. For the *creative arts* subject the set of classification terms may include *opera*, *ballet*, and *Mozart*. The set of terms which best classifies each subject is used to query each collection and the number of times each term occurs in each collection is recorded. Accordingly these results are then used to classify each collection. It can be shown that Collection A is more suitable for finding information about the creative arts than Collection B, and any time a user requests information about the creative arts, Collection A can be returned as the best possible source for information.

As stated above, a collection can be treated as a black box with no prior knowledge of its contents. All that is known is that when some information is sent, some information is returned from it. Based on what is returned some knowledge it gained about its contents. There are two main questions that need to be answered in order to find information about the contents of the black box.

1. Decide what to send to the black box?
2. Decide what to do with the information returned from the black box?

An ontology answers the first question. By using an ontology, we aim to make collection selection more precise. Subject classification terms are used to probe black boxes.

Taxonomy addresses the second question. By transforming the probe term results into a taxonomy, a detailed view of the subjects contained in the black box is achieved.

In collection selection, *query probing* [6, 5] is commonly used to discover the contents of uncooperative collections. Query probing involves sending a set of query terms to a collection and using the results to form conclusions about the collection's content.

3.1 Methods of Collection Selection

The method for evaluating each collection for each query is as follows:

1. Extract "<title>" queries from TREC Topic Queries 51-100.
2. Convert each query into a set of four third-level Dewey Decimal codes using the Q.U.T. Library catalogue search engine.
3. Convert each Dewey Decimal code into a set of query probe terms taking the ten terms with highest support and confidence values for each Dewey Decimal code from the ontology.
4. Send each set of query probe terms to each of the collections one-by-one using the Zettair [1][8] search engine.
5. Extract the number of results for each query probe term from the Zettair results.
6. Sum the results from Zettair together and use them to rank each collection.

In our method the query probe terms from each subject node of the third level of the taxonomy are extracted. While it was difficult to decide how many classification terms to extract for each subject node, the use of more terms allows better results for collections which have a wider but more shallow coverage of a subject. However these collections may not have as high quality results as ones that provide deeper results for part of a subject. The use of fewer terms would result in better results for collections which have a deeper coverage of some aspects of a subject but poor results for collections which have a wider coverage of a subject. In our experiments the top ten results from the highest confidence and support for each subject node are used.

Once the query probe terms for each subject have been extracted from the IntelliOnto ontology they are sent to each collection. The number of results for each term from each collection is extracted and saved.

Once the query probe terms have been sent to the collection, and the results gathered, the terms need to be grouped into Dewey Decimal subject codes. To calculate the Dewey Decimal subject code results, the sum of the set of terms used to query probe the collection for each Dewey Decimal subject is taken. For example, if ten terms from a subject are used to query probe a collection, the results for each of the ten terms will be added together and this result recorded as the result for this subject code.

The query score for each subject in each collection is the sum of the ten results for each of the ten query probe terms.

4 Related Work

In this work a large scale ontology was built and used for collection selection. Literature related to this ontology based collection selection method is now reviewed.

4.1 Collection Selection

Collection selection is becoming more and more important as the number of collections on the internet increases daily. *Collection selection* is the matching of a set of related collections with an information need. The problems of collection selection have been addressed in previous work such as CORI [4] and GlOSS [14]. CORI assumes the best collections are the ones that contain the most documents related to the query. GlOSS uses a server which contains all the relevant information of other collections. Users query GlOSS which then returns an ordered list of the best servers to contact to send the query to. In a comparison of CORI and GlOSS [7] it was found that CORI was the best collection selection method, and

[8] Zettair is a compact open source TREC and HTML search engine from the R.M.I.T. University.

that a selection of a small number of collections could outperform selecting all the servers and a central index.

Web based collection selection introduces its own set of problems, in that there is usually no direct access to a collections statistics, and that there is rarely cooperation between the collections and the collection broker. Our previous work [19, 21] in web based collection selection used query sampling methods that did not require communication with the broker or metadata about each collection. Singular value decomposition was then used on the results of the queries to select the best collection. These techniques were tested on the INEX collection with satisfactory results. In other work [36], a subject based approach was used for information fusion and was found to be promising and efficient. In [20] a short preview of the work presented in this paper was presented.

Si et. al. [31] present a web based modification of CORI called *ReDDE* which performs as well as or better than CORI by using a collection size estimate to supplement selection. They introduce an collection size estimation technique which is more efficient than in other estimation techniques such as the capture-recapture method [24].

Hawking et al [16] presented a method which used both centralised and distributed collection selection techniques. They also made use of anchor text to extract information on collections that have not been indexed.

Si et. al. [32] presented a method for minimising the poor quality results returned by collections which have not implemented good information retrieval methods. By including the retrieval performance of each collection in the collection ranking, this problem can be reduced. A method for approximating the retrieval effectiveness of a collection, known as RUM, was presented. The RUM method was compared to CORI and outperformed CORI in all the experiments conducted.

A common problem with traditional collection selection techniques are that they require communication between the search broker and collections, or that they need topical organisation. In this paper we presented a form of collection which does not need communication between the search broker and collections, and does not need topical organisation.

4.2 Ontology Learning

There is a growing body of work covering automatic and semi-automatic ontology learning. Automatic ontology learning has emerged as a separate entity from other ontology research, drawing from data mining, machine learning and psychology. However, automatic ontology learning is still very difficult to achieve other than in very specialised domains. We will briefly summarize some of the key research to date.

Maedche et. al. [27] presents methods for semi-automatically extracting ontologies from domain text. This includes methods for determining the measure of relationship between terms and phrases. Some ontology mining algorithms have been mentioned in [28, 26], which are the discoveries of the *backbone taxonomy* and the non-taxonomic relation.

Esposito et al. [9] provided semi-automatic ontology learning based methods for transforming raw text into a computer readable representation, enabling a computer to learn a language from training examples.

Faure et. al. [10] claims to have built a machine learning clustering system which learns subcategorization frames of verbs and ontologies from unstructured technical natural language texts. Unfortunately, in this example the methods were only tested within a single limited domain of cooking recipes which is itself highly structured (ie ingredients and cooking methods are fields common to all recipes).

Buitelaar [2] selected 126 classification types and used WordNet as an ontology to assign almost forty thousand polysemic noun terms to one or more types in an automatically generated ontology. Each term could be disambiguated by what set of categories it belonged to or is excluded from. These groupings could then be used to tag corpora to aid automatic processing of data.

Suryanto et. al. [34] applied ontology learning to an existing well structured ontology allowing rapid extraction of rules. Kietz et. al. [18] applied semi-automatic ontology learning tools to a company intranet environment where natural language was mined for information.

Li et. al. [22, 23] presented a method of automatic ontology learning and refinement which can be used to model web user information needs. Stojanovic [33] used an ontology to refine search queries by removing term ambiguity. Queries were taken and mapped to their neighborhoods in a controlled vocabulary, then the neighborhoods were presented to the user for assessment. Gauch [35] uses hierarchal weighted ontologies to create a personalised user profile and to assist web browsing. The ontologies are used to classify web pages and user browsing habits into different categories, and the user profiles are matched to spidered web paged. Gandon [12] provided methods for managing distributed knowledge and assisting corporate activities by using ontologies.

The above references all contain examples of ontology generation and ontology learning. However many of the above examples use only a small, domain specific ontology with limited application. In this work we automatically create a large ontology covering hundreds of different domains.

5 Experiment Data

For the experiments, four well known collection selection testbeds were derived from the TREC Tipster collection ; Trec123-100col-bysource, Trec123-2ldb-60col, Trec123-AP-WSJ-60col, and Trec123-FR-DOE-81col. These testbeds cover a range

of environments, from the base which contains many small collections of the same size (trec123-100col), a mixture of small uniform sizes and two large collections with similar relevant document density (Trec123-2ldb-60col), a mixture of small uniform sizes and two large collections which contain a high concentration of relevant documents (Trec123-AP-WSJ-60col), and a mixture of small uniform sizes and two large collection which contain a low concentration of relevant documents (Trec123-FR-DOE-81col). For a full description of each testbed see Si and Callan [31].

For our queries we take the TREC Topic Queries 51-100 from the Tipster collection. Because the IntelliOnto ontology was computed form the Queensland University of Technology's library catalogue, pilot studies in overlap between TREC queries and the ontology revealed deficiencies in the ontology's coverage. As the primary goal was to assess performance without relying on collection size estimates, only those queries deemed to map suitably into the ontology were employed. The queries from the TREC Topics 51-100 used are 63, 65, 66, 70, 71, 74, 75, 82, 85, 86, 96, 97, and 98. We will cover the full 50 queries in later work. We experimented sending the actual query as both a phrase and as single terms and found that the query performed much better as a phrase than as single terms.

6 Experiment Results

The experiments evaluating the IntelliOnto were done on four testbeds(Section 5). The relevance judgements for each of the TREC Topic Queries 51-100 were taken from TREC website, the file names were *qrels.51-100.disk1.disk2.parts1-5.tar.gz* and *qrels.51-100.disk3.parts1-5.tar.gz*. From these the "ideal" baseline was computed: for each query, collections were ranked on decreasing order of the number of relevant documents they contain.The ideal calculated a baseline for each collection using these relevance judgements. To shed light on the question of collection size estimation, the ReDDE system was used as it employs a well motivated collection ranking algorithm, whereby collection size estimation is a crucial feature.

$$R_k = \frac{\sum_{i=1}^{k} E_i}{\sum_{i=1}^{k} B_i} \quad (3)$$

where B is a baseline ranking and E is the collection selection algorithm ranking. B_i and E_i are the number of relevant documents counted for position i in the ranking. The larger the value R at position k, the better the ranking method E.

The top 20 collections for each testbed were selected. Figure 3 shows the results of the collection selection on the four testbeds. The IntelliOnto system performed best on the trec123-2ldb-60col representative collection, and worst on the trec123-100col-bysource collection. We believe that the reason for this differential in performance is due to collection size; the IntelliOnto method appears to perform better on larger collections. Due to the preliminary nature of this work, we can only speculate as to why this is the case. One possible reason is that the probability of encountering subject classification terms is higher in larger collections thus allowing them to be ranked more effectively. The favourable comparison with ReDDE on the larger collections does lend some support to our hypothesis that effective collection selection is possible without using estimates of collection size. Larger experiments and a more detailed analysis will be needed to bear this out.

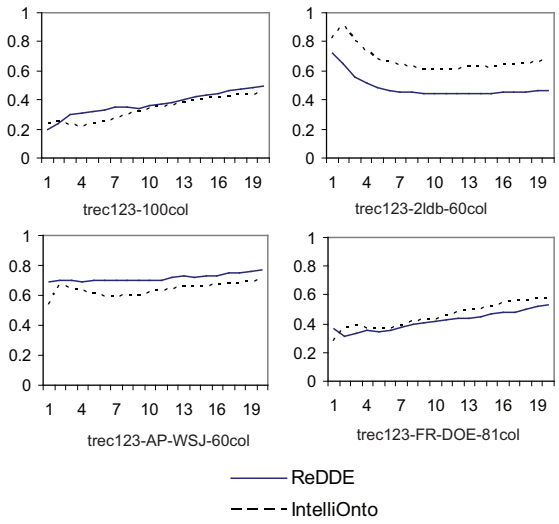


Figure 3: Collection selection accuracy on the testbeds.

7 Summary and Conclusions

This article tackles the collection selection problem from the query side. Queries are enhanced by mapping them to subjects in an ontology; the associated subject classification terms are then employed to retrieve collections. A novel form of ontology based collection selection, IntelliOnto, is introduced. This method was compared to ReDDE, the current state-of-the-art collection selection method. In preliminary experiments, the IntelliOnto method provided encouraging performance on larger collections. Although the research is preliminary, there is some support to the hypothesis that the ontology-based approach mitigates the need for collection size estimates in collection selection. In further work we will fully populate the ontology, and bring in collection size estimates. Work will also focus on using subjects deeper in the ontology with the goal of improving precision. We will also experiment to find how many DDC codes from the search to use for best results, and how many query probe terms to use for best results.

References

[1] B. Billerbeck, A. Cannane, A. Chatteraj, N. Lester, W. Webber, H. E. Williams, J. Yiannis and J. Zo-

bel. RMIT University at TREC 2004. In E. M. Voorhees and L. P. Buckland (editors), *Proceedings Text Retrieval Conference (TREC)*, Gaithersburg, MD, November 2004. National Institute of Standards and Technology Special Publication 500-261.

[2] P. Buitelaar. *CoreLex: Systematic Polysemy and Underspecification*. Ph.D. thesis, Computer Science Department, Brandeis University, 1998.

[3] French J.C. Powell A.L. Callan, J. and M. Connell. The effects of query-based sampling on automatic database selection algorithms. In *Technical Report CMU-LTI-00-162*, Carnegie Mellon University, 2000. Langauage Technologies Institute, School of Computer Science.

[4] J. P. Callan, Z. Lu and W. Bruce Croft. Searching Distributed Collections with Inference Networks . In E. A. Fox, P. Ingwersen and R. Fidel (editors), *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28, Seattle, Washington, 1995. ACM Press.

[5] Jamie Callan, Margaret Connell and Aiqun Du. Automatic discovery of language models for text databases. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 479–490. ACM Press, 1999.

[6] Nicholas Eric Craswell. *Methods for Distributed Information Retrieval*. Ph.D. thesis, The Australian National University, 2001.

[7] Nick Craswell, Peter Bailey and David Hawking. Server selection on the world wide web. In *Proceedings of the fifth ACM conference on Digital libraries, San Antonio, Texas, United States*, pages 37–46. ACM Press, 2000.

[8] Daryl J. D'Souza, James A. Thom and Justin Zobel. A comparison of techniques for selecting text collections. In *Proceedings of the 11th Australasian Database Conference(ADC'2000)*, pages 28–32, Canberra, Australia, 2000.

[9] F. Esposito, S. Ferelli, N. Fanizzi and G. Semeraro. Learning from parsed sentences with INTHELEX. In Claire Cardie, Walter Daelemans, Claire Nédellec and Erik Tjong Kim Sang (editors), *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop, Lisbon, 2000*, pages 194–198. Association for Computational Linguistics, Somerset, New Jersey, 2000.

[10] D. Faure and C. N'edellec. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *In LREC workshop on Adapting lexical and corpus resources to sublanguages and applications, Granada, Spain*, 1998.

[11] James C. French, Allison L. Powell, James P. Callan, Charles L. Viles, Travis Emmitt, Kevin J. Prey and Yun Mou. Comparing the performance of database selection algorithms. In *Research and Development in Information Retrieval*, pages 238–245, 1999.

[12] Fabien Gandon. Agents handling annotation distribution in a corporate semantic web. *Web Intelligence and Agent Systems, IOS Press*, Volume 1, Number 1, pages 23–46, 2003.

[13] Luis Gravano and Héctor García-Molina. Generalizing GlOSS to vector-space databases and broker hierarchies. In *International Conference on Very Large Databases, VLDB*, pages 78–89, 1995.

[14] Luis Gravano, Héctor García-Molina and Anthony Tomasic. GlOSS: text-source discovery over the Internet. *ACM Transactions on Database Systems*, Volume 24, Number 2, pages 229–264, 1999.

[15] David Hawking and Paul Thistlewaite. Methods for information server selection. *ACM Transactions on Information Systems (TOIS)*, Volume 17, Number 1, pages 40–76, 1999.

[16] David Hawking and Paul Thomas. Server selection methods in hybrid portal search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 75–82, New York, NY, USA, 2005. ACM Press.

[17] Xiaohui Tao Richi Nayak John D. King, Yuefeng Li. Mining world knowledge for analysis of search engine content. *Web Intelligence and Agent Systems: An International Journal*, October 2007. Accepted for publication in September 2006.

[18] Joerg-Uwe Kietz, Alexander Maedche and Raphael Volz. A method for semi-automatic ontology acquisition from a corporate intranet. In *Proceedings of EKAW-2000 Workshop "Ontologies and Text", Juan-Les-Pins, France, October 2000*, number 1937 in Springer Lecture Notes in Artificial Intelligence (LNAI), 2000.

[19] John D. King. Deep web collection selection. Master's thesis, School of Software Engineering, Queensland University of Technology, 2003.

[20] John D King. Large scale analysis of search engine content. In *The Fourth International Conference on Active Media Technology, Brisbane, Australia*, Volume 1, page 451 to 453, 2006.

[21] John D. King and Yuefeng Li. Web based collection selection using singular value decomposition. In *IEEE/WIC International Conference on Web Intelligence (WI'03)*, pages 104–110, Halifax, Canada, 2003.

[22] Y. Li and N. Zhong. Capturing evolving patterns for ontology-based web mining. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 256–263, Beijing, China, 2004.

[23] Y. Li and N. Zhong. Mining ontology for automatically acquiring web user information needs. *IEEE Transactions on Knowledge and Data Engineering*, Volume 18, Number 4, pages 554–568, 2006.

[24] King-Lup Liu, Clement T. Yu and Weiyi Meng. Discovering the representative of a search engine. In *CIKM*, pages 652–654, 2002.

[25] Z. Lu, J.P. Callan and W.B. Croft. Applying inference networks to multiple collection searching. Technical Report TR96–42, University of Massachusetts at Amherst. Department of Computer Science, 1996.

[26] A Maedche and S Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, Volume 16(2), pages 72–79, 2001.

[27] Alexander Maedche and Steffen Staab. Discovering conceptual relations from text. In W. Horn (editor), *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*, pages 321–325, 2000.

[28] Alexander Maedche and Steffen Staab. Learning ontologies for the semantic web. In *SemWeb*, 2001.

[29] Weiyi Meng, King-Lup Liu, Clement T. Yu, Wensheng Wu and Naphtali Rishe. Estimating the usefulness of search engines. *15th International Conference on Data Engineering (ICDE'99)*, Volume 1, pages 146–153, 1999.

[30] Milad Shokouhi, Justin Zobel, Falk Scholer and S. M. M. Tahaghoghi. Capturing collection size for distributed non-cooperative retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 316–323, New York, NY, USA, 2006. ACM Press.

[31] L. Si and J. Callan. Relevant document distribution estimation method for resource selection, 2003.

[32] Luo Si and Jamie Callan. Modeling search engine effectiveness for federated search. In *SIGIR*, pages 83–90, 2005.

[33] Nenad Stojanovic. Information-need driven query refinement. *Web Intelligence and Agent Systems, IOS Press*, Volume 3, Number 3, pages 155–170, 2005.

[34] H. Suryanto and P. Compton. Learning classification taxonomies from a classification knowledge based system. In C. Nedellec P. Wiemer-Hastings S. Staab, A. Maedche (editor), *Proceedings of the Workshop on Ontology Learning, 14 Conference on Artificial Intelligence (ECAI'00)*, Berlin, 2000. Conference on Artificial Intelligence (ECAI'00).

[35] Jason Chaffee Susan Gauch and Alexander Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems, IOS Press*, Volume 1, Number 3, pages 219–234, 2003.

[36] Xiaohui Tao, John D King and Yuefeng Li. Information fusion with subject-based information gathering method for intelligent multi-agent models. In *The Seventh International Conference on Information Integration and Web-Based Applications and Services, Kuala Lumpur, Malaysia*, Volume 2, page 861 to 869. iiWAS, 2005.

Document-related Awareness Elements in Synchronous Collaborative Authoring

Gitesh K. Raikundalia *Hao Lan Zhang*

School of Computer Science and Mathematics
Victoria University
PO Box 14428
Melbourne City MC 8001 Australia

Gitesh.Raikundalia@vu.edu.au *haolan@sci.vu.edu.au*

Abstract *Simultaneous collaboration on documents by distributed authors has been supported by numerous synchronous collaborative authoring systems that are widely available. Originally, these tools were found to lack in providing rich enough interaction during authoring. As a result,* group awareness *in collaborative authoring arose as a very important issue in understanding how to provide comprehensive knowledge about other authors and activities they perform upon the document. To promote effectual authoring of documents simultaneously, group awareness is required to allow authors the best possible understanding of others' work on the document.*

This paper reports results about document-related awareness elements *from an empirical and experimental study of group awareness. Awareness elements reflect fundamental awareness information in supporting group awareness. Such results teach us what sort of document-related awareness should be provided for collaborative authoring.*

Keywords Document management, collaborative document authoring, group awareness.

1 Introduction

Real-time, distributed, collaborative writing systems (RDCWS) allow a group of geographically dispersed authors to work on a document simultaneously. Different RDCWS, such as GROVE [1], SASSE [2] and REDUCE [3], have been developed over the last two decades for users to author documents. Although these tools allow two or more authors to work on the same document at the same time, such a tool is not necessarily used to write an entire document (from beginning to end) in one session. Participants could use tools like email or workflow to write parts of a document asynchronously, whilst other parts are written together synchronously using the RDCWS. More importantly, these tools are not used as widely as they could possibly be used. One of the reasons for insufficient usage of RDCWS is that existing RDCWS have not supported the richness and diversity of interaction found face-to-face.

Figure 1 shows a screen capture of *REDUCE*, which was the RDCWS used in this research. The Figure shows two users writing a document with REDUCE. The two background colours highlight the text entered by each of the two users. Essentially, REDUCE can be understood as being a collaborative equivalent of a single-user word processor.

Understanding current, past or even future work on a document is essential for human interaction. People find it simple when face-to-face to gain a sense of awareness about who is present, what are their responsibilities, what they are doing and where they are located. When group members are dispersed, supporting interaction is far more problematic due to different reasons such as limited views or relatively poor communication [4]. Hence, there is a tremendous need for *group awareness* to provide the highest-quality collaborative authoring [5].

Group awareness (which we simply refer to as *awareness*) is defined as "an understanding of the activities of others, which provides a context for your own activity" [6]. Awareness is relevant to group interaction for various reasons, such as facilitating communication, support of coordination [1] and allowing use of "convention" amongst users [5]. In the case of collaborative authoring, users are provided with knowledge about the document and about past, current or future activities other users carry out with the document.

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.

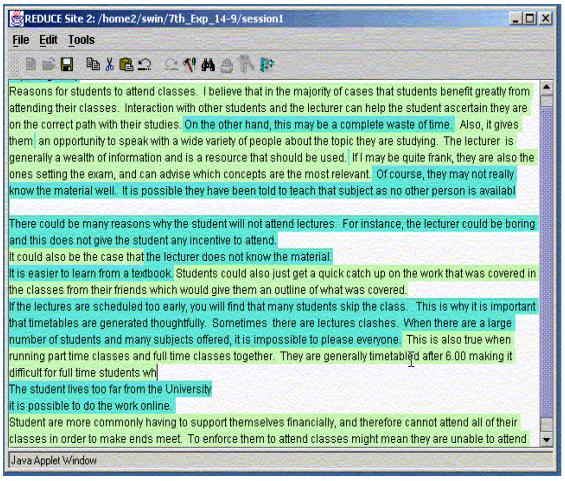


Figure 1: Collaborative authoring ([11])

Awareness elements have been contributed by Gutwin and Greenberg [4]. Awareness elements reflect foundational awareness information needed to support group awareness. Examples of document-related awareness elements include Knowing the parts of a document at which other users are currently looking, or Knowing the parts of a document on which other users are currently working. These elements are types of awareness strongly associated with the document as opposed to an element such as Knowing who is in the workspace which is knowledge of which other authors have joined the authoring session.

It is highly pertinent to study such elements as these elements reflect the information authors require to provide them awareness. This information indicates the support of awareness during collaboration, and therefore reveals the types of functionality provided by *awareness mechanisms*. Awareness mechanisms developed over the years include those such as radar views [7] or modification director [8]. For instance, radar views describe a high-level view of a document, showing where authors are located in the document. Therefore, if from experimental results for collaborative authoring there is support for the element, Knowing the parts of a document on which other users are currently working, it means that a mechanism is required to show where in the document other users are either currently entering, modifying or deleting text.

The aims of this project are to determine:

- which awareness information is relevant for support of group awareness, and
- the relative importance of these types of awareness information.

Gutwin and Greenberg have very usefully prescribed a set of awareness elements in their conceptual framework for workspace awareness. This paper contributes experimental results for document-related awareness elements, which yield findings for awareness support from these results. These findings are applicable to developing novel awareness mechanisms extending the present set available for awareness support. The results are indicative of which awareness information is in comparison more relevant for design of mechanisms than other less relevant awareness information. A further contribution is placed at the end of this paper: using these results to form an explication of how an effective collaborative authoring session can occur. Such an explication can be included because the results teach us about how document-related awareness can be supported in a real-world authoring session.

2 Related work

As mentioned previously, Gutwin and Greenberg proposed group awareness elements. However, we have computed empirical, quantitative results for these elements to determine their relative importance, and therefore, which types of awareness are more pertinent for support.

We have published results from this same experiment that are for *non-document related awareness elements* [9]. Examples of these elements include: Knowing if other users are satisfied with what you have done and Being able to view the list of past actions carried out by a specific user. Such elements are not associated specifically and strongly with work on a document unlike a document-related element such as Knowing the parts of a document at which other users are currently looking. Results for non-document related awareness elements are presented in [9], whereas this paper concentrates on results for document-related awareness elements. A separate experiment with a different number and set of users is reported in [11].

In addition, some major awareness mechanisms that provide awareness support in collaborative authoring are covered in related work. Each of these mechanisms in its basic essence is providing information related to some awareness element(s).

Radar views [7] are awareness mechanisms that support a high-level view of a document. A radar view displays the places in a document where all authors are working. One of the problems with radar views is the gap existing between low-level details and the global structure of a document.

To overcome the problem of this gap, the extended radar view [8] has been developed. The extended radar view uses the over-the-shoulder view, which allows content to be more readable than in the radar view, yet is lower resolution than in the actual document. Authors are able to pick up much content from the over-the-shoulder view without having to use the actual document itself.

Telepointers [10] are a mechanism allowing several cursors—a cursor for each author—to be used within a document. Telepointers assist in showing all

		Experimental sessions											
		E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E 10**	**E 11**	**E 12**
Silence first	**CW**				T1/ T2		T1/ T2						
	DP			T5/ T6					T5/ T6				
	BS		T3/ T4			T3/ T4							
Verbalisation first	**CW**	T1/ T2									T1/ T2		
	DP							T5/ T6					T5/ T6
	BS									T3/ T4		T3/ T4	

Table 1: REDUCE experimental sessions

the parts of a document that all users are working on concurrently.

Multi-user scrollbars [7] use the well-known scrolling facility of a window to allow an author to see where in a document other authors are working. In this case, there are multiple scroll bars—one bar for each user. A user uses the scroll bar to scroll up and down the document whilst viewing where other users are working on the document.

Modification director [8] is a mechanism that tracks changes in a document. However, since authors are working simultaneously on the document, these changes need to be notified to authors as they occur in time. Details that are notified to authors include the text being modified, the page where the modification takes place, the time elapsed since the modification took place, the type of deletion (if the modification was a deletion) that occurred, etc.

Various other awareness mechanisms have been developed over the years, hence, the above list is by no means exhaustive.

3 Experimental design

The RDCWS used in our laboratory-based experiment was the REDUCE editor [3], which basically contains almost no awareness support. The Swinburne Usability Laboratory of Swinburne University of Technology, Melbourne, Australia, was where the usability experiment was carried out.

24 experimental subjects were placed into twelve pairs and worked on three writing tasks. The writing tasks were *creative writing* (CW) (e.g., writing short essays from scratch), *document preparation* (DP) (e.g., writing a software manual) and *brainstorming* (BS) (i.e., idea generation). The two reasons for using these particular categories are that these categories:

1. reflect a variety of collaborative authoring tasks
2. require different styles of collaboration

The tasks of CW, DP and BS were carried out by the twelve pairs in the following way: 4 pairs worked on CW, 4 pairs worked on DP and 4 pairs worked on BS. The Appendix shows the actual tasks used in experiments. The combination of tasks used by a pair in a session is conveyed by Table 1. As an example, two DP tasks are allocated to the pair in session E8. The first DP task given to the E8 pair is task T5 (T5 is shown in the "Experimental tasks" part of the Appendix) where there is no communication via telephone. The second task given to this pair is task T6 where communication is allowed during collaboration.

Each individual subject of a pair was placed in one of two separate subject rooms. As would occur in real-world distributed collaboration, subjects were unable to view each other from their rooms. From an observation room where notes could be taken, a research assistant observed each pair. The two-and-a-half hour session in which each pair participated included the following activities.

Training subjects in REDUCE (1/2 hour). Subjects learnt how to use REDUCE to author collaboratively.

Experiment (1 hour): Subjects worked together on one task for half-an-hour without verbal communication (silence) and for half-an-hour on another task with verbal communication (verbalisation). Conducting the experiments with and without verbal communication allowed the possibility of determining problems users had when there was only silence and the methods they adopted to address this difficulty.

Questionnaire and interview (1 hour): Each subject filled in a semi-structured questionnaire containing nineteen six-point scale (closed-ended) questions and thirteen open-ended questions. Closed-ended questions were asked to discover if subjects

Awareness elements	Mean	Std dev.	Awareness elements	Mean	Std dev.
In the case of nonverbal communication, having a communication tool that supports communication between users	4.50	0.60	Being able to view the list of past actions carried out by a specific user	3.72	0.98
Knowing the tasks for which other users are responsible	4.33	0.87	Knowing what actions other users are going to take in the future	3.70	1.02
Being able to comment on what other users have done	4.30	0.75	Knowing if other users can know what you have been doing	3.68	0.99
Knowing the parts of a document on which other users are currently working	4.22	1.00	Knowing to what extent a portion of a document has been completed	3.64	1.05
Knowing what actions other users are currently taking	4.08	1.02	Knowing which part of a document at which other users are currently looking	3.36	1.14
Having voice communication	4.04	1.22	Having video communication	3.30	1.29
Knowing who is in the workspace	3.91	0.97	Knowing how long other users have been in the workspace	3.00	1.14
Knowing if other users are satisfied with what you have done	3.91	1.11	Knowing how much time has elapsed since other users have used REDUCE	2.78	1.09
Seeing the position of other users' cursors on the screen	3.78	1.24	Knowing where other users are physically located	2.04	1.30
Knowing to what extent you have completed your work compared to the extent others have completed their work	3.74	1.02			

Table 2: Results for awareness elements ([9])

believed certain types of awareness were relatively important or unimportant for supporting collaborative authoring. The results for awareness elements in this paper were determined from these questions. Open-ended questions discovered from subjects the mechanisms they believed were useful for supporting group awareness. The open-ended questions are not relevant to this paper. Each subject could further clarify these mechanisms during the one-on-one interview held by the research assistant whilst filling in the questionnaire. Audiotape was used to record interviews with subjects for assistance with data analysis.

4 Results of analysis

We now present the results of analysis of the close-ended questions. The results assist to differentiate the necessity of different awareness elements. This paper focuses only on results for document-related elements. The mean and standard deviation of all close-ended questions are computed and the distribution of responses for each question is constructed. Each closed-ended question corresponds to an awareness element. Therefore, the mean and distribution of responses of a question reflect the relative importance of an awareness element.

The awareness elements are sorted by their means in Table 2. The elements which are document-related and for which results are reported here include:

- Knowing the parts of a document on which other users are currently working
- Knowing to what extent you have completed your work compared to the extent others have completed their work
- Knowing to what extent a portion of a document has been completed
- Knowing the parts of a document at which other users are currently looking

An awareness mechanism is suggested as worthwhile to develop if, on balance, there is a larger proportion of subjects favouring the element from the results. A

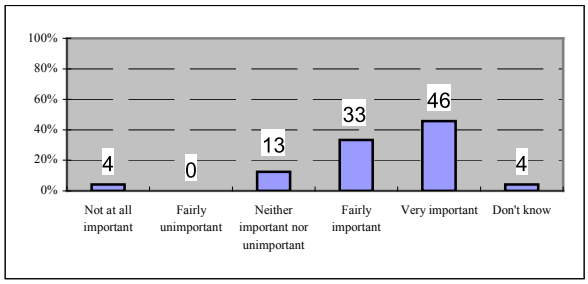


Figure 2: Knowing the parts of a document on which other users are currently working

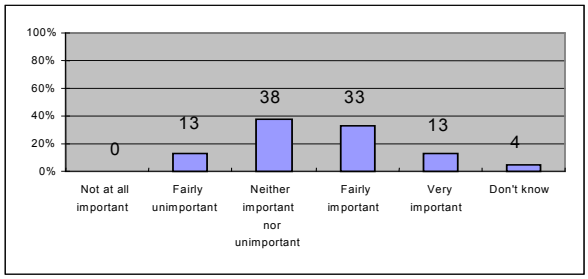


Figure 3: Knowing to what extent you have completed your work compared to the extent other users have completed their work

potential mechanism has to be evaluated experimentally to determine if it is truly effective before the mechanism is used in real-world authoring.

4.1 Parts on which other users work

As can be seen from Figure 2, almost half of the subjects found that knowing all the parts of a document that other authors are working on, at the current point in time, to be a very pertinent form of awareness. Also, a third of the subjects found this awareness to be reasonably important. Therefore, mechanisms that provide awareness of where others are currently working are viewed as important. Telepointers, radar views and multi-user scrollbars provide this awareness. The issue here is not that awareness mechanisms showing where users are working do not exist and need development, but that given the shortcomings of these mechanisms, there is much more scope for providing such awareness. Indeed a mechanism for providing this awareness was discovered from this experiment and is shown in the Appendix. We have called the mechanism *Structure-based Multi-page View*.

4.2 Completing your work compared to others' work

Figure 3 shows the distribution of responses when subjects were asked about the importance of knowing how much of their work on the document has been done relative to how much others' work has been done on the document. Such a characteristic could be measured in different ways; what is intended by such an awareness element is the actual concept of being knowledgeable of how far other users have reached in completing their work compared to you.

The distribution shows that 46% of participants believe there is high or reasonable importance in having this knowledge. Thus, in a collaborative authoring session, half of the subjects could be expected to use, to varying degrees, a software mechanism providing this type of awareness. Such a mechanism is not expected to be one of the more highly utilised and popular mechanisms according to our results. However, for the purpose of enhancing users' experience as much as possible and supporting them flexibly, such a mechanism would be provided, even if 60% – 70% of users do not make use of it. An aim of general research on group awareness for collaborative authoring is to offer a "palette" of awareness mechanisms from which users choose appropriate mechanisms to provide them with awareness—all users will not use exactly the same mechanisms during an authoring session.

4.3 Portion of document completed

As can be seen from Figure 4, half the subjects believed that being aware of how much has been completed of each of the different parts of a document is of importance. Thus, from their

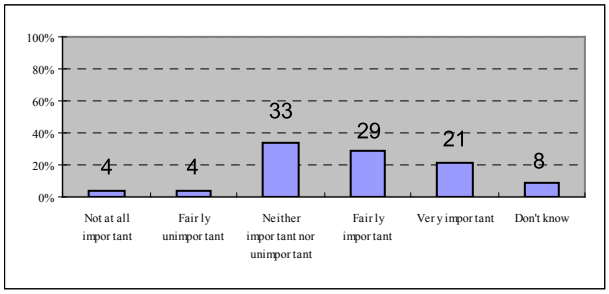


Figure 4: Knowing to what extent a portion of a document has been completed

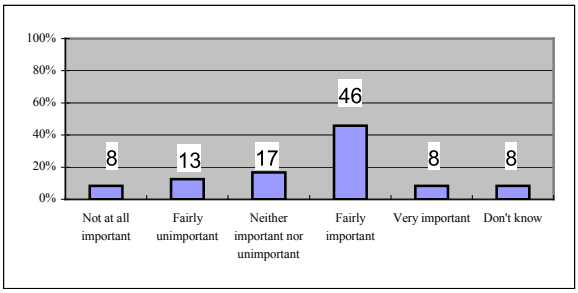


Figure 5: Knowing the parts of a document at which other users are currently looking

experience in the experiment, half the other subjects did not provide the authors with a convincing response that there was need for such awareness. This means that half the subjects would like the availability of a software mechanism providing updated progress of completion of parts of the document. The definition of a portion or part of a document is open and depends on the document being created. Although given that a fifth of the subjects found such awareness to be highly important in collaborative authoring, discovery of a software mechanism that provides this awareness is worth pursuing.

4.4 Where others are currently looking

Clearly, when users are collaborating on a document, they will be viewing somewhere inside the document. A user could be viewing the portion of the document that they are working on (e.g., where this user is adding text, removing text, etc.) or could be viewing somewhere that they are not working on, such as some other user's work. Thus, there can be a difference between a user's working area and their viewing area.

Figure 5 shows that almost half the subjects found it fairly important to know, at a given point in time, where the other subjects were looking in the document. Therefore, there was not an overwhelming need for awareness support to know where other users are viewing currently, but more than half of all subjects found reasonable or high importance in having such support. The conclusion here is similar to the conclusions for the previous two awareness elements: investigate what sort of software mechanisms would provide this awareness and develop them for the palette of mechanisms for users; however, anticipate them to be less used than mechanisms supporting the first awareness element in this section.

5 A group awareness-supported collaborative authoring session

The above histograms and associated discussion provide results informing us about group awareness during authoring. Therefore, from these results, we obtain an idea of the awareness needed for an effectual collaborative authoring session. In other words, an effectual session involves the use of document-related awareness covered in the remainder of this section. Other general forms of awareness, as represented by the elements in Table 2, are not covered here since they do not relate to this paper.

Many users find during the session that they need instant and easy access to information about which user is working on which part of the document. There could be a host of reasons for desiring such information during the session. For instance, an author wants to refer to a section being written by another author, and the first author needs to find out if the second author has finished writing the section yet. Another example is where an author wants to know if another author has finally completed a particular section of work on the document. The first author is interested in know whether or not the second author's section is finished as yet. This type of information is undoubtedly fundamental when a group is working as a team on the contents of a document. Current mechanisms such as telepointers and radar views already provide this awareness, thus, the authors' results confirm the relevance of these mechanisms. However, developing new mechanisms that provide this awareness in more relevant ways is also justifiable.

Some users feel they want to know the progress of others' work on the document. These users wish to know how far from completion are other users' work on various parts of the document. It may be that user *A* cannot work on part *a* of a document until user *B* has completed part *b* of the document as completion of part *b* is a pre-requisite to commencement of work on part *a*. Also, user *C* needs to apply content from part *d*, worked on by user *D*, to their own part, part *c*. Thus, user *C* needs to know when user *D* completes part *d*. An awareness mechanism for representing this type of progress is used by users *A* and *C* during authoring.

Additionally, a mechanism is used by some users to determine how far they have progressed in their work on the document in comparison to how far others have progressed on the document. Used at different points by some of the users during co-authoring, such a mechanism assists a user either to:

- feel more confident and fulfilled when the user is ahead in their contribution compared to others, or
- know they are progressing satisfactorily in their contribution compared to others, or
- feel concerned if their contribution is progressing more slowly compared to others.

Some users may want to know where other users are viewing within the document. A user may want to know if other users, who have some right to evaluate their contribution to the document, are viewing their contribution. In another case, user *A* may be able to determine if another user, user *B*, is viewing their contribution for a long period of time. This would notify user *A* that user *B* is taking interest in user *A*'s contribution, and that user *B* may be about to discuss the contribution with user *A* or even go ahead and amend the contribution. If user *A* is aware that user *B* is spending a great deal of time viewing user *A*'s contribution, this will be indicative of the seriousness (whether it be positive or negative) behind user *A*'s contribution.

6 Conclusion

The awareness element of Knowing the parts of a document on which other users are currently working received the strongest support. This suggests that not only are mechanisms providing this type of document-related awareness important for providing to users, but that they will probably be the most used document-related mechanisms during authoring. The other remaining elements whose results were presented in this paper received enough support to justify discovery of mechanisms providing these types of awareness (since few mechanisms currently provide these types of awareness).

A mechanism that supports the element, Knowing to what extent you have completed your work compared to the extent others have completed their work, appears to provide emotional support rather than support of the work itself. However, even in face-to-face interaction, we can see that satisfaction, disappointment and other emotional effects are experienced through awareness of others during work.

Acknowledgements This project has been funded by Victoria University Discovery Research Grant Scheme 2004. Many thanks to Prof. Yun Yang and Dr Minh Tran of Swinburne University of Technology and to John Craick, Manager of Swinburne Usability Laboratory.

References

[1] C. Ellis, S. Gibbs and G. Rein. Groupware: some issues and experiences. *Communications of the ACM,* Volume 34, Number 1, pages 39–58, 1991.

[2] R. Baecker, D. Nastos, I. Posner and K. Mawby. The user-centred iterative design of collaborative writing software. In *InterCHI'93*, pages 399–405, Amsterdam, 24 – 29 April 1993.

[3] Y. Yang, C. Sun, Y. Zhang and X. Jia. Real-time cooperative editing on the Internet. *IEEE* Internet *Computing*, Volume 4, Number 1, pages 18–25, 2000.

[4] C. Gutwin and S. Greenberg. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work*, Volume 11, Number 3-4, pages 411–446, 2002.

[5] J. Grudin. Groupware and social dynamics: eight challenges for developers. *Communications of the ACM*, Volume 37, Number 1, pages 92–105, 1994.

[6] P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *1992 ACM Conference on Computer Supported Cooperative Work*, pages 107–114, Toronto, Canada, November 1992.

[7] C. Gutwin, M. Roseman and S. Greenberg. A usability study of awareness widgets in a shared workspace groupware system. In *1996 ACM Computer-Supported Cooperative Work*, pages 258–267, Boston, USA, November 1996.

[8] M. Tran, Y. Yang and G. K. Raikundalia. Extended Radar View and Modification Director: awareness mechanisms for synchronous collaborative authoring. In *Seventh Australasian User Interface Conference*, pages 45–52, Hobart, Australia, January 2006.

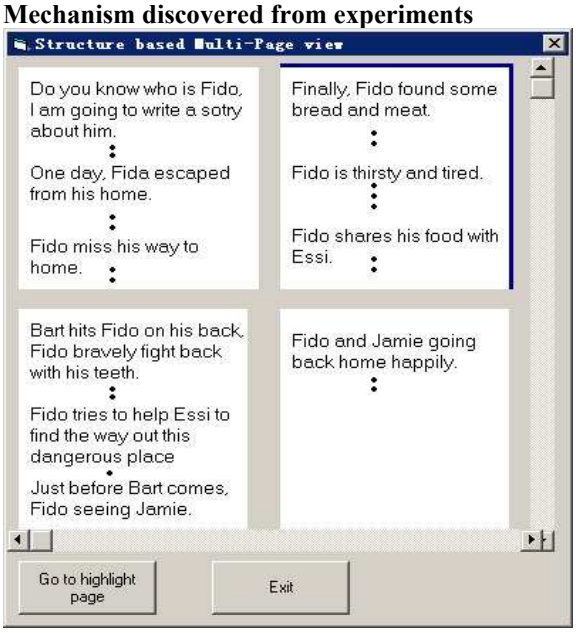
[9] G. K. Raikundalia and H. L. Zhang. Experimental findings for awareness elements in real-time, distributed, collaborative authoring. In *OZCHI04*, University of Wollongong, Australia, November 2004.

[10] S. Greenberg, C. Gutwin and M. Roseman. Semantic telepointers for groupware, In *Sixth Australian Conference on Computer–Human Interaction*, pages 54–61, Hamilton, NZ, November 1996.

[11] M. Tran, G. K. Raikundalia and Y. Yang. What are you looking at? Newest findings from an empirical study of group awareness. In *APCHI 2004*, pages 491–500, Roturua, NZ, June – July 2004.

Appendix

Mechanism discovered from experiments



Experimental tasks

Creative Writing

T1: "Fido is a dog living in Melbourne and owned by a boy, Jamie. Write a fictional story about the adventures of Fido."

T2: "Write a fictional story about the various events that occur in a sports team playing in a particular match. For instance, a soccer team or a cricket team or a basketball team, etc. playing a particular match."

Brainstorming

T3: "Stress affects people in modern life. There are clearly many different ways of escaping the stress and difficulties of modern life. Write down and explain various ways of reducing stress."

T4: "Write down different problems and difficulties that you feel occur when being taught in an educational setting (e.g., university lecture, workshop carried out in a company, etc.)"

Document Preparation

T5: "Write a research paper on an agreed topic with the other participant."

T6: "Write a manual or guide about REDUCE. This manual/guide must instruct and teach the reader how to use REDUCE."

Questionnaire – six-point closed-ended questions

A response to each one of the close-ended questions below will be one of the following six points, as indicated by the experimental subjects in the questionnaire:

1 – Not at all important
2 – Fairly unimportant
3 – Neither important nor unimportant
4 – Fairly important
5 – Very important
6 – Don't know

Close-ended questions

1. Knowing who is in the workspace
2. Knowing the tasks for which other users are responsible
3. Knowing how much time has elapsed since other users have used REDUCE
4. Knowing where other users are physically located
5. Knowing how long other users have been in the workspace
6. Being able to view the list of past actions carried out by a specific user
7. Knowing the parts of a document on which other users are currently working
8. Knowing the parts of a document at which other users are currently looking
9. Knowing what actions other users are going to take in the future
10. Knowing what actions other users are currently taking
11. Seeing the position of other users' cursors on the screen
12. Knowing to what extent you have completed your work compared to the extent others have completed their work
13. Knowing to what extent a portion of a document has been completed
14. Knowing if other users can know what you have been doing
15. Being able to comment on what other users have done
16. Knowing if other users are satisfied with what you have done
17. Having voice communication
18. Having video communication
19. In the case of nonverbal communication, having a communication tool that supports communication between users

A Sequence Based Recommender System for Learning Resources

Dean Cummins *Kalina Yacef* *Irena Koprinska*

School of Information Technologies
University of Sydney
NSW 2042 Australia

(dcummins, kalina, irena) @it.usyd.edu.au

Abstract *This paper presents a novel approach for recommending sequences of resources for users to view based on previous user feedback. It considers the order in which resources are viewed to be important in delivering the next set of suggestions and tries to learn these dependencies from users' ratings. Although we describe our approach in the context of e-learning, it can be applied to other domains where ordering is important. We also propose a novel algorithm for learning the dependencies between the resources. Preliminary results are encouraging: they show that, after a threshold in quantity of feedback, our algorithm provides better results than standard collaborative filtering.*

Keywords Digital Libraries, Document Management, Information Retrieval

1 Introduction

Online learning resources can be very convenient to help users learn or practice a topic. Like any other resources that are part of a very large set which can potentially grow infinitely, such as movies or books, users appreciate some guidance in selecting the appropriate resource. There are two major ways to provide this guidance: one is based on *annotation* and the other on *collaborative filtering*. The first one typically relies on resource content metadata and requires the resource structure to follow some standards. Thus, a significant human effort is required to annotate and structure the resources. The second way relies on previous users' ratings to make recommendations, and this is where our research falls in. In particular, in this paper we study whether users' ratings could be used to achieve reusability of learning resources.

There are already a number of techniques and algorithms to build collaborative recommender systems (two famous systems are MovieLens [16] and Amazon [2]). However, there are two interesting aspects that distinguish learning resources from other items:

(i) the order in which the learning resources are seen is important: there is no point in seeing the difficult material without seeing the basics first;

(ii) there is an end to the process of seeing learning resources. Learners typically see a small, finite number of resources for a given topic, whereas there is no end to the process of seeing movies or reading books.

These two aspects make the recommendation problem interesting and novel. We propose a sequence-based recommender approach to make suggestions to learners on what resources to use in order to learn about a given topic. Importantly, our approach aims at suggesting one or more learning paths. Learners seek new resources on the open web and add them to the pool of resources, which then grows over time. As the resources are procured from the open web and are referenced simply by a URL, the goal of reusability is achieved.

The next section discusses related research on resource reusability in online learning. Section 3 then formally defines the problem and presents our approach, highlighting the issues and challenges that lay ahead. Section 4 introduces the dependency learner algorithm, on which our approach relies on. Section 5 presents the experiments conducted to evaluate the algorithm and discusses the results. Finally Section 6 concludes the paper.

2 Background and existing work

In order to promote the reusability and identification of useful educational content on the web, the term *learning object* was coined [18]. The broad idea is centered on classifying suitable resources as learning objects and having them available in purpose built systems for teachers and learners, such as a Learning Object Repository [17]. These repositories aim to be a searchable catalog, enabling the sharing and reusibility of learning objects.

There are many definitions of learning objects and they are not all consistent [18]. The IEEE Learning Technology Standards Committee (LTSC) [10] defines learning objects as *any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning*. Others define it with a much smaller scale, such as a *digital learning resource that facilitates a single learning objective which may be reused in a different context* [15]. Regardless, an analysis of online repositories [7, 13] clearly shows

learning objects to fit the first definition, or more specifically *any digital resource that can be reused to support learning* [21].

While all potentially useful resources available on the web fit these general definitions of learning objects, they are not in these learning repositories. It takes large amounts of human effort to identify, annotate and place these resources into these repositories. To perform this for all resources on the web is a gigantic and infeasible task for a small group of humans to do manually. This means that many potentially useful resources are left untouched if instructors and learners rely on these repositories alone. A learning system which does not require resources to exist in purpose-built repositories and does not require annotation would clearly be more generic and have a larger number of resources to choose from.

2.1 Annotation/metadata approaches

For learning objects to be shared and reused, current approaches involve adding some structure or metadata to the learning objects so that both humans and machines can understand them. This is similar to the semantic web vision in which information is annotated with well defined meaning so that computers and humans can better work in cooperation [3]. Mohan [15] argues that learning objects themselves should play a greater role in the search process and should intelligently interact with learning systems to provide instruction. This would allow intelligent search agents to identify pools of learning objects that are suitable for specific instructional goals. For this to be realised much work has focused on creating and implementing metadata standards and supporting ontologies.

Current popular metadata standards include Dublin Core [5] and IEEE's Learning Object Metadata (LOM) [11]. These are supported by a large number of metadata creation software, repositories and learning systems. The reality is that metadata creation is a lengthy, boring and tedious process with an undesirable high human cost involved [4]. Consequently elements are often overlooked or used incorrectly and inconsistently [8]. It was further identified that to promote semantic interoperability, common vocabularies or ontologies are required. However this is currently not the case [15, 19, 4].

Without consistently and extensively annotated learning objects, current learning systems cannot interoperate with each other and make effective use of the vast amount of available resources. The goal of achieving common standards and interoperability of all available learning objects therefore does not appear to be realistically achievable in the near future.

2.2 Collaborative filtering approaches

Another approach commonly used for sharing and reusing documents is to use social or collaborative filtering. For instance, recommendation systems aim to provide the user with choices or recommendations based on how users rated the resources, hence can even provide personalised recommendations. Current popular applications of these systems include MovieLens [16] and Amazon [2]. More recently we have seen collaborative filtering applied to electronic documents such as document searches [12] and research papers recommendations [14, 20]. However these approaches are focused on recommending single, unrelated resources whereas we are interested in recommending resources which may have implicit orderings within them.

For example, if we implemented a recommender system for resources using traditional techniques, resources would be suggested based on previous ratings by all learners and learners similar to the current learner, without taking into account sequencing between resources. More specifically, consider three resources A, B and C. A and B are two resources which deal with basic concepts for some topic, and C covers more advanced material. A traditional recommendation list will potentially include all three items, (A, B, C), ranked by the popularity of previous ratings given by learners. A learner may pick any of the three resources with the impression that top ranked resources are more useful and liked by other learners.

However, we wish to take into account that there may be implicit dependencies between resources. As C deals with more advanced material, it would be appropriate to recommend resources A and B to the learner before recommending C, regardless of whether C has been rated higher overall. Once the learner has seen A or B and found them useful then C should be recommended, thus presenting the resources in an order that takes sequencing into account.

We can also extend this recommendation by making available to the learner full paths of resources such as $(A \rightarrow C, B \rightarrow C)$ ahead of time. This provides the learner with a more structured environment that is similar to traditional custom learning systems but using a recommendation based approach. Learners are then able to visualise suggested learning paths, not just lists of resources.

3 The approach

In contrast with using metadata or standard collaborative filtering, our approach learns and extracts the dependencies that exist in the learning resources based on user ratings. Our approach does not rely on human annotation of learning resources and aims at discovering not only popular resources but also popular paths. Once we have mined these potential dependencies, we combine them into a dependency graph that can then be used by an intelligent recommender system to suggest resources as shown in Figure 1.

In our proposed system, the learner is suggested a set of learning paths (i.e. sequences of learning resources as identified in the dependency graph), has

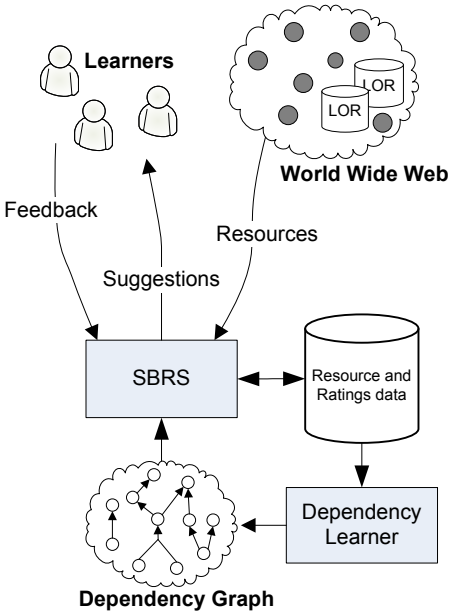


Figure 1: Proposed Sequence Based Recommender System

the possibility to view the other resources as well as to pick new resources from the open web. After viewing each resource, the learner is asked to provide a rating (feedback), which is added to the resource rating database. The system has a dependency learner, described in Section 4, which builds and maintains the dependency graph of resources.

3.1 Proposed Sequence Based Recommender System

We describe the main sections of our proposed sequence-based recommender system.

Growing the pool of resources. To take advantage of the vast amount of resources on the world wide web, we propose that learners and instructions be given the ability to add resources into the system that they believe will be useful for the given topic, similar to social bookmarking sites such as Del.ico.ous [6]. This will likely grow a pool of relevant resources with a high degree of quality as it has already been hand picked by a human. Naturally, spam and irrelevant resources may be added to this pool, however if these are not selected and are consistently rated poorly by learners they can be filtered out. We believe this to be a great strength of this approach and will allow the system to evolve over time as the number of resources increases. The system can also link into existing learning object repositories such as [7, 13] and select resources by the appropriate categories. While this may introduce resources that are not relevant for the given topic, these will be soon identified through the continuous feedback by learners interacting with the system.

Obtaining the feedback. When learners are presented with resources, they will be required, or greatly encouraged, to provide short, instant feedback in the form of rating. For the dependency extraction to work, it is required to know if the resource was useful or if it relies on content beyond what the learner believes he/she should know and as such, the feedback needs to reflect this. A star *usefulness* rating will perhaps be sufficient and this can be converted into a utility value as expected by the dependency extraction algorithm.

Mining the dependencies. Given a set of learner ratings, we need to extract any potential dependencies. We have developed an algorithm for mining a ratings based dataset to extract potential dependencies. Our approach has been inspired by the data mining problems Association Rule Mining and Sequential Pattern Analysis and is described later in Section 4.

Generating intelligent suggestions. Once the dependencies have been extracted and merged into a graph, the system can then make suggestions based on:

- Resources the learner has seen
- Resources other learners have liked
- The dependency graph

The dependency graph drives the suggestion process as this is the core component that produces the learning path for the learner.

The following section describes a scenario of how the system works on a small example.

3.2 Sample scenario of recommendations

We provide a description on how we propose the sequence based recommender system will recommend resources to a learner. For our sample scenario, we use the dependency graph as shown in Figure 3, which is simple yet contains different types of dependencies (AND, OR dependencies as well as independant resources).

Our pool of resources contains eight resources which are labeled A to H. Each edge in this graph represents a dependency between resources. For example, E depends on either D or C. A special case is the dependency for C which depends on both A and B, but not in any particular order. Each dependency represents what the learner should have seen in order to find the current resource useful. All of these resources are targeted at a particular topic.

A learner who wishes to learn about this topic logs into the system. Initially, the learner has not seen any resources and so the system will select suitable resources for the learner to begin with. The dependency graph is consulted and resources at the leaves, A, B and D are suggested to the learner. We also indicate

Seen	$\emptyset$
Suggested	$A, B, D\ (C, E)$
Other	F, G, H

(a) Iteration 1

Seen	B
Suggested	$A\ (C, E)$
Other	D, F, G, H

(b) Iteration 2

Seen	B, G
Suggested	$A\ (C, E)$
Other	D, F, H

(c) Iteration 3

Seen	B, G, A
Suggested	$C\ (E)$
Other	F, H

(d) Iteration 4

Seen	B, G, A, C
Suggested	E
Other	F, H

(e) Iteration 5

Seen	B, G, A, C, E
Suggested	$\emptyset$
Other	F, H

(f) Iteration 6

Figure 2: A trace of a learner's interactions with the system

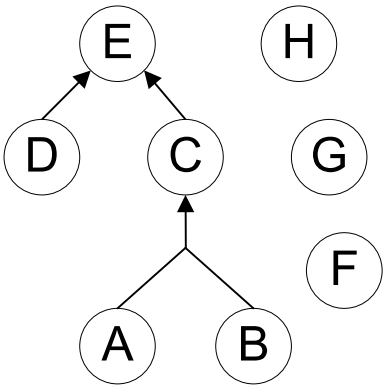


Figure 3: A Dependency Graph

resources that the learner should not see at this point in time, C and E, next to these suggestions in brackets. Finally, any resource that is not connected to the graph is added to an *other* list, so they are not hidden from the user. Figure 2(a) shows the current state of the suggestions for the user.

The learner selects resource B out of the suggested resources. As the dependency graph contains a dependency $\{A, B\} \rightarrow C$, it will suggest A next as B is thought to complement A. This is reflected in Figure 2(b). Figure 2(c) shows that the learner selected G, which does not change the suggestions. Once the learner has seen the suggested resource A (Figure 2(d)), C is selected (Figure 2(e)) and finally, E (Figure 2(f)).

The scenario described shows how we can leverage the dependency graph to present the learner resources they can follow in a suitable sequence. As this is a recommender system and it requires input from its users to make better suggestions over time, it is important for the users to have access to the full set of resources, not just the ones the system believes should be recommended. Over time, as ratings are collected from learners, the system should be able to learn and solidify the relationships between the resources.

For example, it might be the case that G is a resource that is only useful if seen after E as it contains advanced material which requires the knowledge taught in E. It also might be the case that D is not a very useful resource or is unrelated to the topic at hand and thus should be removed from the system. After a number of learners have used the system and provided feedback showing that D is not a very useful resource and that H is useful if viewed after E, the dependency graph should be modified to take the form of Figure 4.

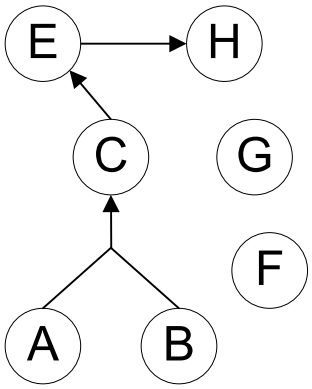


Figure 4: Modified Dependency Graph

Thus this approach differs from traditional recommender systems in that we take sequencing into account. It also differs from other learning based systems in that it works on the feedback (ratings) obtained from the learner, not through metadata records or static learning paths.

We will next describe the important pillar of this approach, the algorithm that mines the dependencies in the learner ratings of resources.

4 Dependency learner algorithm

We describe the main ideas behind our algorithm and how it is used to extract dependencies from datasets. While the algorithm we have implemented is based on this approach, it is similar to Association Rule Mining [1] in that it extracts dependencies satisfying a minimum support and confidence threshold. This makes it more robust so it can handle incomplete datasets (datasets which may not have all the information necessary to extract all dependencies with reasonable confidence) and noise. For the purpose of this paper, we focus on describing a simpler algorithm that extracts dependencies from a consistent dataset with no noise.

ID	Feedback
1	A_+, B_+, C_+
2	A_+, C_-, B_+
3	B_-, A_+, C_-
4	B_-, C_-, A_+
5	C_-, A_+, B_+
6	C_-, B_-, A_+

Table 1: Sample dataset

The task Given a dataset of sequences (records), where each element is an item-utility pair, the goal is to find the dependencies between the items.

For example, given the dataset in Table 1 which contains 6 sequences and 3 items $\{A, B, C\}$, we want to find the dependencies between these items. For the e-learning domain, the items corresponds to resources and the utility values to ratings. Each rating is either a positive (+) or negative (-) rating as denoted by the subscripts in the dataset. For example, if record 1 in the dataset corresponded to the ratings history of a learner, the learner will have viewed resources A, B and C in this order and rated them all positively. The learned dependencies can be represented as a graph where the nodes corresponds to items and the links to dependencies.

4.1 Algorithm rationale

The main idea of the algorithm is to take into account the items rated positively before each positive or negatively rated item in each sequence. We illustrate three different cases for some item i.

1. Any positively rated items that appear before a positively rated i *may collectively be* a dependency for i. For example, in record 1, C has been rated positively thus A or B, both A and B or neither may be a dependency for C.

2. Any positively rated items that appear before a negatively rated i implies that these are not dependencies for i. For example, in record 2 C was given a negative rating. A appears before C with a positive rating which implies that the existence of A is not enough to result in a positive C. There is not enough evidence that C depends on A.

3. Any negatively rated item j that appears before i is ignored. This is because j may have been rated negatively as *its* dependencies did not appear before it. Record 3 has a negatively rated item B appearing before a negative C. We can not exclude B from being a dependency of C as it might be the case that a positive A needs to occur before B to result in a positive B. This situation is actually supported by record 1.

4.2 Algorithm Description

The algorithm consists of the following steps:

For each item $i \in I$, the set of items:

Step 1. Create two *projected datasets* P_{i+} and P_{i-} for the positively and negatively rated items, i_+ and i_-, respectively. A projected dataset for item i is the set of positive items that occur before i in each sequence of the source dataset.

Step 2. Remove from P_{i+}, itemsets in P_{i-} as they do not impact on the ratings of i, creating the potential dependency set D_i.

Step 3. Find the smallest set of itemsets from D_i which describes the dependencies.

Finally the identified dependencies are merged into a dependency graph. This is a directed graph with each node corresponding to an item and each edge to a dependency. The graph is created such that the items on each path from a leaf to some item i is contained within the dependency set D_i.

4.3 Example

In this section, we apply the dependency learning algorithm on the dataset in Table 1.

Step 1. The first step is to take each item and find the items with a positive rating that preceded them in each record. This is done for both positively and negatively rated items. For example, the first record contains a positive B and there is only one positive item, A, preceding it. Thus considering only record 1, $P_{B+} = \{A\}$. This relationship can be used to imply a dependency between A and B, $(A \rightarrow B)$. Note, that if in a subsequent record, we also find the $(B \rightarrow A)$ dependency, then this will cancel out the previous one resulting in no dependency between A and B.

The project datasets for the three items, A, B and C are shown in Figure 5.

For our example:

$P_{A+} = \{\emptyset\}$, $P_{A-} = \{\emptyset\}$

$P_{B+} = \{A, A, A\}$, $P_{B-} = \{\emptyset\}$

$P_{C+} = \{(A, B)\}$, $P_{C-} = \{A, A\}$

Step 2. Next, we work out D_i which is defined as P_{i+} with all itemsets of P_{i-} removed. As i is not dependent on itemsets in P_{i-}, then by removing these from P_{i+} we only consider the items that may be dependent on i.

For example, if $P_{A+} = \{B, C, D\}$, $P_{A-} = \{B, C\}$, then $D_A = \{D\}$ as this is the only set of items that matters as A can not be dependent on $\{B, C\}$.

For our example:

$D_A = \{\emptyset\} - \{\emptyset\} = \{\emptyset\}$

$D_B = \{A, A, A\} - \{\emptyset\} = \{A, A, A\}$

$D_C = \{(A, B)\} - \{A, A\} = \{B\}$

Step 3. Now we take the smallest subset of the potential dependencies, D_i. This involves finding the

ID	P_{A+}	P_{A-}
1	$\emptyset$	
2	$\emptyset$	
3	$\emptyset$	
4	$\emptyset$	
5	$\emptyset$	
6	$\emptyset$	

(a) Projected datasets for A

ID	P_{B+}	P_{B-}
1	A	
2	A	
3		$\emptyset$
4		$\emptyset$
5	A	
6		$\emptyset$

(b) Projected datasets for B

ID	P_{C+}	P_{C-}
1	$\{A, B\}$	
2		A
3		A
4		$\emptyset$
5		$\emptyset$
6		$\emptyset$

(c) Projected datasets for C

Figure 5: Projected datasets for the dataset in Table 1

smallest set of items which cover the full set of dependencies for the given item. Given any sequence X and Y if X is contained within Y, we can remove Y.

For example, given that the current item is A, and $X = \{B, C\}$ and $Y = \{B, C, D\}$ where $\{X, Y\} \in D_A$, we can remove Y as X is contained within Y. This means that A will be positive if positive items B and C exist before it in sequence. As this subsequence exists in both X and Y, we simply keep the smaller of the two and discard the other.

For our example:

$D_A = \{\emptyset\}$

$D_B = \{A\}$

$D_C = \{B\}$

Finally, we build the graph of dependencies. Every item i is dependent on each item within Y given $Y \in D_i$. (Note that if there are multiple dependencies, $|D_i| > 1$, i will be positive if at least one of the dependencies exists before i in a sequence.)

For our example:

$A \rightarrow B$

$B \rightarrow C$

Thus the dependency graph is $A \rightarrow B \rightarrow C$.

5 Experiments

Evaluating our approach and proposed system involves a live user experiment [9] which is timely to perform. Before conducting this expensive stage in our research, we first needed to thoroughly evaluate our algorithm in a simulated environment in order to 1) assess whether our algorithm can learn a dependency graph based on simulated user ratings and 2) gain understanding on the impact of numerous factors on its effectiveness and efficiency. We have conducted a comprehensive set of experiments of which we will present some results.

We have developed a simulation framework in which an abstract version of the resource suggestion problem can be modeled. The simulation framework requires a dependency graph to be defined which reflects the *real* dependencies of the resources. We then pass a number of artificial learners through the simulation which provides a rating for each resource depending on what the artificial learner has seen. Each dependency graph contains a set of *goal* nodes in which the artificial learner aims to reach - this is akin to finishing a topic by looking at all suitable resources until the topic is learnt. Once a goal node is reached, the artificial learner stops requesting resources and exits the simulation. Artificial learners select suggested resources 85% of the time, otherwise selects a resource from the *other* category. (Refer to Figure 2 for the three types of recommendations our system presents). Currently we deal with a consistent scenario, such that if an artificial learner has seen and liked all of the dependencies for some resource r it will rate r as being positive. We have performed two main types of experiments - those that explore how well the dependency graph can be learnt and those that compared different recommendation techniques. The selected experiments presented in this paper were run 100 times with the results averaged.

The first set of experiments we ran aimed to explore our algorithms ability to learn dependencies. For each run of the experiment, we compared the learnt dependency graph with the real dependency graph, identifying correct, missing and incorrect edges. Results are presented for three basic types of dependency graphs - a linear path, a binary bottom up tree and a binary top down tree as shown in Figure 6.

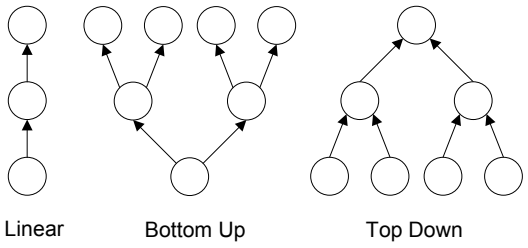


Figure 6: Graph structures experimented with.

The linear path contains a number of resources which all need to be viewed in a specific (linear) order to reach the goal. A binary bottom up tree contains a single leaf (root) node, branching out and ultimately leading to one of many goal nodes. A binary top down tree contains many leaf nodes leading to a single goal node. Figure 7 and Figure 8 show the recall and precision results for each learnt dependency graph. In this experiment, the dependency graph is relearnt

every 5 artificial learners. The linear graph contains 10 nodes, while the binary trees have a depth of 3 (thus 4 nodes need to be viewed to reach a goal, however there are 8 different paths to this goal).

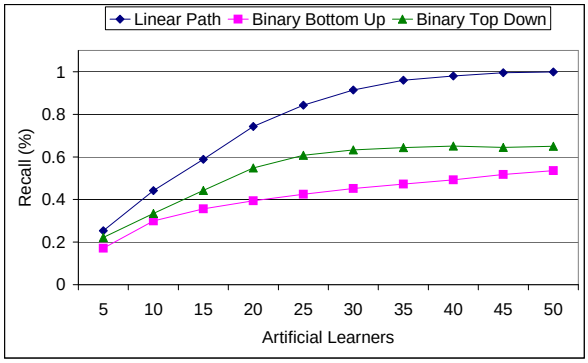


Figure 7: Recall (proportion of correctly learnt edges out of all correct edges) for three different dependency graphs.

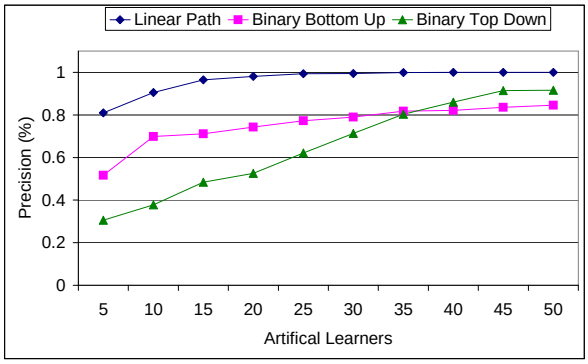


Figure 8: Precision (proportion of correct edges out of all learnt edges) for three different dependency graphs.

The graphs show that the learnt dependency graph is improving over time as more data is collected, particularly in the trivial linear path case. When multiple paths to a goal node exist, as in the case for the binary trees, it is harder to learn the full graph as once a path to a goal node has been identified, other paths are not generally learnt. This is a side effect of the artificial simulation - we propose that when selecting resources, real learners will use additional contextual information such as a resources title and description in addition to the recommendations made by the system. This is one critical aspect that requires future research.

The other main set of experiments we performed focused on comparing different recommendation techniques. We present some results based on a binary bottom up tree (described previously) with a depth of 3. Figure 9 shows how many resources an artificial learner needed to select before reaching its goal. This is something we would like to minimise, to avoid presenting redundant or useless resources to the learner. Figure 9 shows the percentage of resources that the artificial learner rated positive, out of all resources it selected. The system should avoid recommending (by explicitly discouraging) resources which should not be selected as its dependencies have not yet been selected. We see that randomly suggesting resources delivers poor results, due to the unlikely nature that suggestions will be made in a suitable order. Furthermore, suggesting resources based on its popularity alone (top rated) tends to result in nearly all resources being selected in order to reach a goal. (Resources at the lower level of a dependency graph will be rated positive more often than those at the top). Supporting the recommendation process via the learnt dependency graph clearly outperforms both random and top rated as it suggests a minimal number of resources that allows the learner to reach its goal.

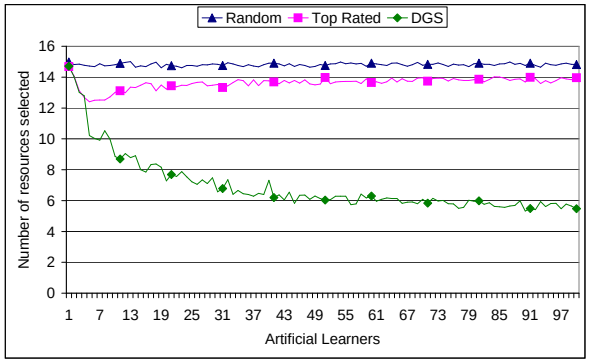


Figure 9: Number of resources selected in order to reach the goal.

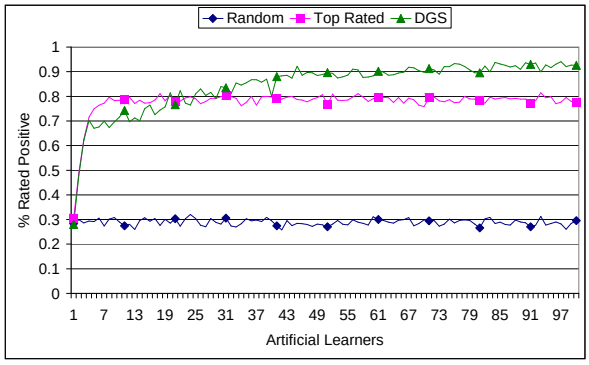


Figure 10: Percentage of resources rated positive out of those selected.

Thus, the simulation results show that the proposed approach can learn dependencies, and paves the way for a trial with real users.

6 Conclusion

We have presented an approach for recommending sequences of learning resources based on previous users' ratings. The novelty of this approach resides in the creation and use of a dependency learner algorithm. This work is still in progress and avenues for future work are numerous. We have implemented the algorithm and ran tests on simulated data which show very encouraging results. Whilst the e-learning context gave us the motivation for this work, it clearly can be applied to

any other domains where the order in which users see resources is important.

References

[1] Rakesh Agrawal, Tomasz Imielinski and Arun Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM Press.

[2] Amazon. http://www.amazon.com/, Accessed 24th April 2006.

[3] T. Berners-Lee, J. Hendler and O. Lassila. The semantic web. *Scientific American*, Volume 284, pages 34–43, 2001.

[4] C. Brooks and G. McCalla. Towards flexible learning object metadata. In *Int. J. Cont. Engineering Education and Lifelong Learning*, Volume 16, pages 50–63, 2006.

[5] Dublin Core. http://dublincore.org/, Accessed 11th Oct 2006.

[6] del.icio.us. http://del.icio.us/, Accessed 11th Oct 2006.

[7] Edna. http://www.edna.edu.au/, Accessed 11th Oct 2006.

[8] N. Frizen. Final report on the "international lom survey", 09 2004.

[9] J.L. Herlocker, J.A. Konstan, L.G. Terveen and J.T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, Volume 22, Number 1, pages 5–53, 2004.

[10] IEEE. http://www.ieee.org/, Accessed 11th Oct 2006.

[11] IEEE-LSTC. The learning object metadata standard. http://ieeeltsc.org/wg12LOM/lomDescription, Accessed 25th April 2006.

[12] S. Jung, J. Kim and J.L. Herlocker. Applying collaborative filtering for efficient document search. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI04)*, pages 640–643, 2004.

[13] MERLOT. http://www.merlot.org/, Accessed 11th Oct 2006.

[14] S. Middleton, D. De Roure and N. Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. In *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture*, pages 100–107, New York, NY, USA, 2001. ACM Press.

[15] P. Mohan and C. Brooks. Learning objects on the semantic web. In *Advanced Learning Technologies, 2003. Proceedings. The 3rd IEEE International Conference on*, pages 195–199, 2003.

[16] MovieLens. http://movielens.umn.edu/, Accessed 24th April 2006.

[17] F. Neven and E. Duval. Reusable Learning Objects: a Survey of LOM-Based Repositories. 2002.

[18] P. Polsani. Use and abuse of reusable learning objects. *Journal of Digital Information*, Volume 3, Number 4, February 2003.

[19] L.P. Santacruz-Valencia, I. Aedo, A. Navarro and C.D. Kloos. An ontology-based mechanism for assembling learning objects. In *Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/ELearning on Telecommunications Workshop*, pages 472–477, 2005.

[20] T. Ya Tang and G. McCalla. Mining implicit ratings for focused collaborative filtering for paper recommendations. In *UM03' Workshop on User and Group models for web-based adaptive collaborative environments*, 2003.

[21] D. Wiley. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. 2002.

Information Access Efficiency: a Measure and a Case Study

Shijian Lu and Cécile Paris

CSIRO ICT Centre
Locked Bag 17,
North Ryde NSW 1670 Australia

(Shijian.lu, Cecile.paris)@csiro.au

Abstract *One of the advantages we claim for information synthesis and aggregation is that it results in more efficient information access for end users, especially when the relevant information comes from multiple heterogeneous data sources. Although that claim is plausible, it has not been verified by any qualitative studies. It is even unclear how one would quantify the efficiency of information access. In this paper, we propose a measure and then report on a study to identify the information access efficiency gain of a potential application involving information synthesis and aggregation.*

Keywords information access efficiency, information aggregation, information access time and speed, information relevance.

1. Introduction

In the age of information, the competitive edge of an organisation is often defined not by how much information it possesses, but by how efficiently it can be accessed and how it will empower its staff to make good decisions quickly by accessing relevant information. What is relevant information is highly context sensitive. It could depend on the nature of the organisation, users' functional role or task.

In CSIRO, we are developing an information delivery platform (called the Myriad Platform) [5]. It enables the development of applications which automatically generate tailored documents, aggregating and synthetising information from a number of heterogeneous data sources. The effectiveness of tailoring has often been reported in the literature (e.g., in tutoring applications [3] or in medical applications [1]). But there has been no study as to the effectiveness of aggregation in supporting the information seeking task, independently of the tailoring. One of the advantages we claim for information synthesis and aggregation is that it is faster for end users to access relevant information, which could come from multiple heterogeneous data sources. Although intuitive and plausible, this claim has not been verified to our knowledge by any quantitative studies, performed independently of the tailoring. It is even unclear how one would quantify information access efficiency to be able to compare various approaches to accessing information. In this paper, we define a way to measure information efficiency, grounding a qualitative description on some measurable physical entities. We also present a specific study aiming at validating our claim that information aggregation and synthesis improves information access efficiency.

Our investigation is undertaken in the context of a specific potential application for presenting an end user with aggregated information. This application is set within a research organisation, and the scenario is as follows: in such an organisation, staff members often need to know about each other to a certain extent in order to effectively collaborate on various matters. They typically require several pieces of information about a fellow staff member. Currently, different information about a staff member is scattered in various data sources and needs to be accessed via different applications or access points. Our hypothetical application is a staff information web site where information aggregated from various data sources is immediately provided. This is a very simple scenario. Yet it gives us a concrete scenario allowing us to investigate issues around information efficiency: how much efficiency in information access would we gain by developing this hypothetical application?

To start with, then, we first identified what information staff are interested in when looking up a fellow staff member. Based on this information, we recorded exactly what it takes to find those information items within the existing organisation infrastructure, asking questions such as: how many applications/tools are needed? And what kinds of interactions are required? We mapped these interaction types to interaction time, and proposed some metrics to quantify the time required to obtain all this information using the current tools and an information efficiency measure. We then postulated that our hypothetical application would be able to aggregate all the information and present it in one virtual page to the information seeker. We compared the information access measures for the current situation and the hypothetical application to derive a measure of information access efficiency gain.

This paper is organised as follows. We first describe how to identify what information is relevant in the application context. Our approach for measuring information access efficiency is then elaborated. Finally, we conclude by discussing some implications of our study.

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.

2. Gathering information about fellow staff members

In the first part of our study, we used questionnaires to gather staff's opinions as to what information they typically look for. The questionnaire contained three types of questions. The first type was concerned with information relevance. Here, we ask subjects to list the top ten information categories they seek about a fellow staff member. The second type of questions was about current staff information seeking patterns (e.g., how often are you seeking staff information?, what means do you use? and how long does it normally take). The third type of questions was to do with subjects' demographical information. In total, there were 12 questions in the questionnaire. The questionnaire was distributed to 22 randomly selected staff members via email. The subjects were asked to reply within a week's time. One day before the initial due date, a reminder was sent to those subjects who had not yet replied. In the end, ten responses were received.

After receiving the responses, data was collected according to specific question. Regarding to the question of what staff information you are interested to know, over twenty categories of information were registered, including regarding someone's hobbies, and someone's colleagues. Clearly, different people will seek different information. As tailoring is not the focus of this paper, we will not explore those issues further. Instead, we focus on the common subset of staff information categories that emerged from the data set.

Figure 1 shows the top 11 relevant information categories. Over 40% of subjects agree on the top seven categories of staff information, namely, phone number, publications, email address, expertise, (physical) address, experience, and qualification. It is interesting to note that among the top 7 categories of staff information, two of them, expertise and experience, are not explicitly retained by the organisation. The remaining five categories (which we now consider our top 5 categories) come from different data sources in the organisation as shown in Table 1. For example, staff qualification information comes from the human resource system, while publications are stored in the organisation's library database. We realise that, sometimes, the information might already be aggregated in someone's web page; there are, however, wide variations in the information included in staff webpages. Furthermore, a webpage already constitutes a document aggregating information. For our study, we thus explicitly went back to the organisational data sources to find the information, in alignment with the fact that most staff members reported that they use the intranet to find information about someone else.

Staff information categories	**Data sources**
Phone number	Staff look up web page
Publications	Library database
Email	Staff look up web page
Physical address	Intranet
Qualification	Human resource system

Table 1. Data sources for the top five categories of staff information

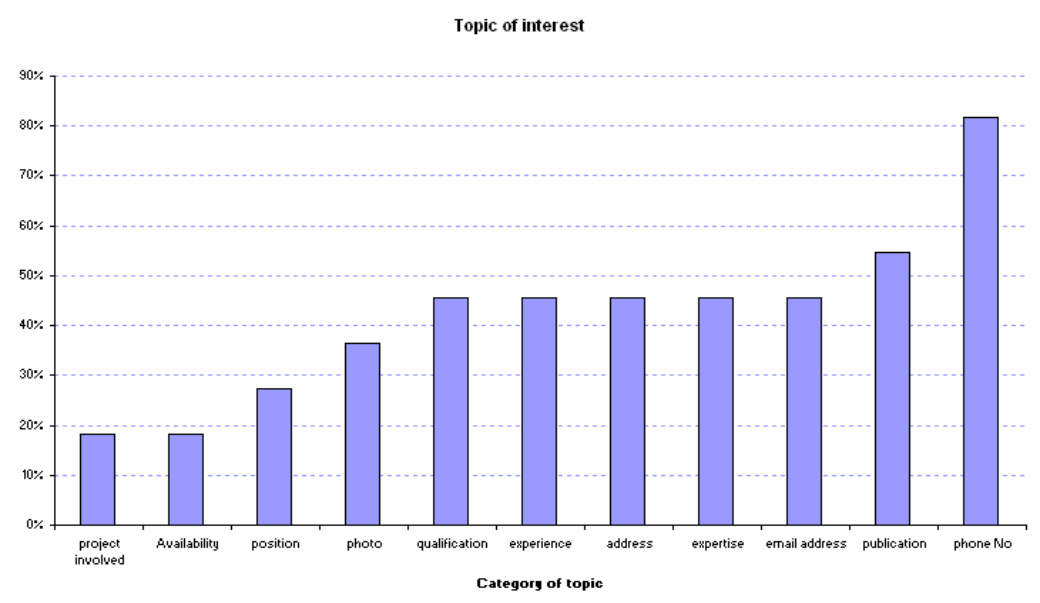


Figure 1. Top 11 staff information categories.

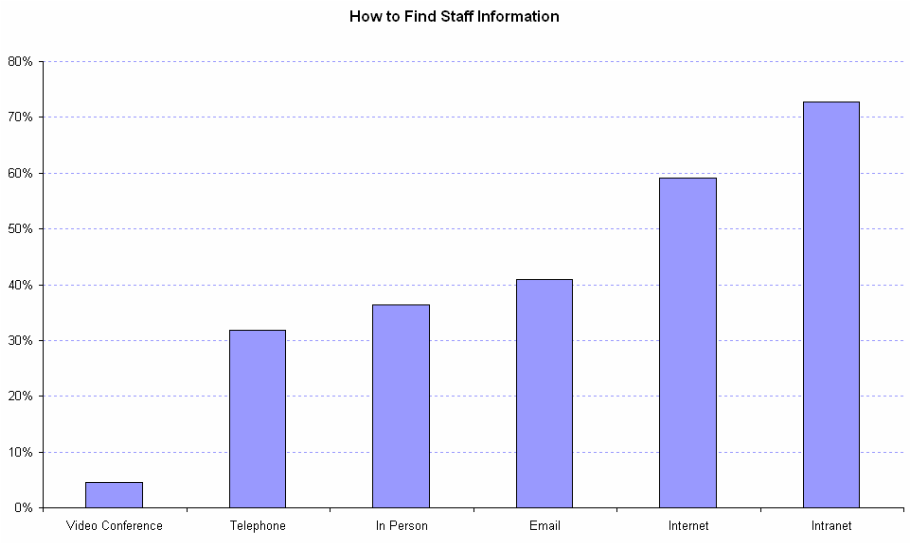


Figure 2. How people go about finding staff information

We briefly summarise some of the patterns that emerged. Overall, a lot of people seek staff information daily or monthly. Most people spend in the order of few minutes for a given time, and they felt that they succeeded in obtaining the information they wanted most of the time. On the issue of how people go about finding staff information, it is probably no surprise to find that the intranet is the preferred means followed by the internet and email (see Figure 2). However, the fact that more people opted for in person than telephone is interesting. We will not elaborate further on these issues as it is not the focus of this paper.

3.Information access efficiency: defining metrics

In this section, we present our approach for quantitatively measuring information access efficiency. In this approach, we differentiate two information accessing modes: inter-item *independent* and inter-item *dependent* accessing mode. Inter-tem independent accessing mode refers to the case when any of one information item is accessed alone. Inter-item dependent access takes into account the fact that items of information to be retrieved are not necessarily independent of each other. For example, they may be obtained using the same application, in which case the time to launch the application should be counted only once. In this case, however, the access point might need to be changed. We will simply refer to the two modes as independent access and dependent access for short.

We then introduce the notion of information access time and speed. The access time of an information item is defined as the time required performing the interaction steps required for accessing it. Under the independent access mode, the access time is the sum of application launch time and other interaction time. Under the dependent access mode, the access time for an information item is the time lapsed between accessing two information items. Under the dependent access mode, the access time for an information item is the time required to access a second item of information once one item has been retrieved.

Access speed is defined as the number of items accessible within a unit of time. Using this definition, average speed can be calculated for the two information accessing modes. The average independent access speed provides a measure of how efficient an application is when accessing any single item, while the average dependent access speed provides a measure of how efficient an application is when any item is accessed after other items. Access speed is then used as a measure of information access efficiency. Our hypothetical information aggregation system is then compared with the current intranet based system using these metrics.

3.1. Establishing steps required to access information

In this part of our study, we used the top five categories of staff information identified in our first experiment as a starting point. For each category of information, we recorded what was needed to be done to find that information: for example, how many tools or applications must be used; how many access points one must go through; how many links need to be selected to get the right information. That is performed for both independent access and dependent access modes. Since most users prefer the intranet, we used the intranet as our choice of information access method.

Information category	load application	key in text	switch access point	click/select hyper link or menu item	press action button or Enter key
Phone number	1	1	0	2	1
Publication	1	1	0	6	1
Email	1	1	0	2	1
Physical address	1	1	1	5	1
Qualification	1	1	0	2	1

Table 2. Number of interaction in the independent access case

3.1.1. Independent access interaction steps

As mentioned earlier, inter-item independent accessing mode refers to the case when any of one information item is accessed alone. Independent access interaction steps for an information item are all the interaction steps required to access the information item in the independent access mode. Taking the phone number as an example, it is accessible with the following steps:

(P1) start the web browser and go to the intranet;
- organisation's homepage appears;

(P2) select the *staff* menu item from the *support* menu ;
- *staff* page appears;

(P3) click the *staff lookup* hyper link from the *Staff* page;
- *staff lookup* page appears;

(P4) key in the last name of the sought after staff member, and

(P5) press *start search* button;
- the phone numbers together with the full names for all staffs with the same given last name are displayed.

After we recorded all the interaction steps required to obtain each information category, we classified them into different types, as each interaction step type requires a different amount of time to execute. We used the following four types: load application, key in text, click/select hyper link/menu item, and press action button.

- **load application:** when an application is launched;
- **key in text:** when text needs to be typed into a text field;
- **switch access point:** when switching to a different web page without hyper link support;
- **click/select hyper link or menu item:** when clicking/selecting a hyper link or menu item;
- **press action button or Enter key:** when pressing an action button or the Enter key.

For each information category, we count how many interaction steps falls into the different interaction types. Table 2 provides a summary of the number of steps required for each interaction step type for the top five information categories if they are accessed independently from each other.

For the phone number example, the number of interactions required are 1, 1, 0, 2, and 1 for interaction type load application, key in text, click/select hyper link/menu item, and press action button, respectively. The 2 interactions required for click/select hyper link/menu item refers to steps (P2) and (P3) above. This is summarised in Table 4 below.

Interaction type	Interaction step
load application	P1
key in text	P4
switch access point	NA
click/select hyper link or menu item	P2, P3
press action button or Enter key	P5

Table 4. Mapping between interaction step and interaction type for accessing phone number in the independent mode

Information category	Load application	key in text	switch access point	click/select hyper link or menu item	press action button or Enter key
Phone number	1	1	0	2	1
Publication	0	1	1	6	1
Email	0	0	0	0	0
Physical address	0	1	2	5	1
Qualification	0	1	1	2	1

Table 3. Number of interactions in the dependent access case

3.1.2. Dependent access interaction steps

Inter-item dependent access takes into account the fact that items of information to be retrieved are not necessarily independent of each other. Dependent access interaction steps for an information item are all the interaction steps required to access the information item, in the dependent access mode. Taking the qualification as an example, it is accessible with the following steps after the phone number has been accessed:

(Q1) Switch to the qualification access point;
- Human resource home page appears;

(Q2) Key in staff ID, and

(Q3) Press the Enter key;
- Staff information page appears;

(Q4) Click on the personal hyper link;
- Sub-personal hyper link appears;

(Q5) Click on qualification hyper link;
- Description of qualification appears.

Similarly, the dependent interaction steps for publication, email and physical address can articulated.

Table 4[1] provides a summary of the number of steps required for each interaction type for the four information items if they are accessed after phone number has been accessed.

3.2. Defining Access Time

In order to measure information access efficiency, we first would like to quantify information access time. To that end, we leverage on research in the field of psychology of human computer interaction. In terms of interaction time, three levels of interaction have been proposed [2]: *psychological moment*, *unprepared response* or *immediate behaviour* [4], and *unit task*. *Psychological moment* is the finest level of interaction at roughly 0.1 seconds. An action and a stimulus event that occurs within 0.1 seconds will seem to exhibit cause and effect relations. *Immediate behaviour* is the intermediate level of interaction at about 1 second. Events that happen in this time frame happen too quickly for the user to respond unless prepared. *Unit task* is the coarse level of interaction ranging from 5 to 30 seconds. This is the typical pace of elementary interaction cycle in interactive systems. An example is the time for a routine interaction with an interactive text editor.

Now, if we can reasonably link our interaction type in last section with the levels of interaction, then we can calculate access time for an information category. Intuitively, the interaction types of **press action button or Enter key** and **click/select hyper link or menu item** corresponds to *psychological moment* and *immediate behaviour*, respectively. For **load application, key in name,** and **switch access point,** they all correspond to the interaction level of *unit task.* However, we think the time designated to *unit task*, namely, 5 to 30 seconds, is too coarse grained for our purpose. So, we introduce three finer levels of distinction to *unit task*: *fine unit task*, at roughly 5 seconds; *intermediate unit task*, at roughly 10 seconds; and *coarse unit task*, at roughly 30 seconds. With this new extension, **key in text** and **switch access point** would correspond to *fine unit task*, and **load application** to *intermediate unit task* (Table 5).

Interaction level		Interaction type	Time interval (sec.)
1	*Psychological moment*	press action button or Enter key	0.1
2	*Immediate behaviour*	click/select hyper link or menu item	1
3	*fine unit task*	key in text, switch access point	5
4	*intermediate unit task*	load application	10
5	*coarse unit task*		30

Table 5. Correspondence between interaction levels and interaction types

Using the information provided in Table 5 above, the access time for an information item may be defined as the sum of the products between required time and number of interaction levels for accessing that item.

$$t(x) = \sum_{i=1}^{5} T_i \, f_i(x) \qquad (1)$$

Where, x -- information item, such as phone number and publications;

$t(x)$ -- access time for information item x;

i -- level of interaction;

T_i -- time required for interaction level i;

$f_i(x)$ -- number of interactions at level i required for accessing item x.

3.2.1. Independent access time

Independent access time is the time required to access an information item in the independent access mode. It can be calculated with formula (1) by replacing $f_i(x)$ with the number of interactions at level i required for accessing item x in the independent mode. For example, the independent access time for

[1] At inter-item dependent access mode, email address will be obtained when accessing phone number. Therefore, no more interaction is required.

information item phone number can be calculated by combining the data provided in Table 5 and first row in Table 2 as follows.

$$t(phone\_no.) = 0.1\times1 + 2\times1 + 5\times1 + 1\times10 = 17.1 \quad (2)$$

Likewise, the independent access time for the rest of the top 5 information items can be calculated. They are shown in Table 6.

Information category	Independent access time (sec)
Phone number	17.1
Publication	21.1
Email	17.1
Physical address	25.1
Qualification	17.1
total	97.5

Table 6. Independent access time for the top five staff information categories

3.2.2. Dependent access time

Dependent access time is the time required to access an information item in the dependent access mode. It can be calculated with formula (1) by replacing $f_i(x)$ with the number of interactions at level i required for accessing item x in the dependent mode. The dependent access time for the top 5 information items can be calculated by using the data provided in Table 5 and Table 4. They are shown in Table 7.

Information category	Dependent access time (sec)
Phone number	17.1
Publication	16.1
Email	0
Physical address	20.1
Qualification	12.1
total	65.4

Table 7. Dependent access time for the top five staff information categories

3.3. Defining Information Access Speed

The information access efficiency of an application/system can be measured by the average information access speed. The higher the speed, the more efficient the system. The information access speed can be defined as the number of information items accessible within a unit of time.

The average information access speed $\bar{v}$ can be calculated with formula (3).

$$\bar{v} = \frac{n}{\sum_{j=1}^{n} t_j} \quad (3)$$

Where, n -- the total number of information items;

t_j -- the access time for the j[th] information item.

3.3.1. Independent access speed

The item independent access speed is the average number of information items accessible within a unit of time under the independent access mode. It can be calculated with formula (3) by replacing t_j with the independent access time for the j[th] information item.

Applying the data shown in Table 6 to formula (3), the average information independent access speed $\overline{v_{in}}$ for the system in which users simply have to use the intranet to gain access to the top five information items is calculated as follows:

$$\overline{v_{in}} = \frac{5}{97.5} \approx 0.051 \text{ (items per second)} \approx 3 \text{ (items per minute)} \quad (4)$$

3.3.2. Dependent access speed

The item dependent access speed is the average number of information items accessible within a unit of time under the dependent access mode. It can be calculated with formula (3) by replacing t_j with the dependent access time for the j[th] information item.

Applying the data shown in Table 7 to formula (3), the average information dependent access speed $\overline{v_{de}}$ for the system in which users simply have to use the intranet to gain access to the top five information items is calculated below:

$$\overline{v_{de}} = \frac{5}{65.4} \approx 0.076 \text{ (items per second)} \approx 4.6 \text{ (items per minute)} \quad (5)$$

3.3.3. Information access efficiency

With the information access speed defined in the preceding section, it is possible to evaluate information aggregation systems in terms of information access efficiency. We can thus consider our hypothetical system, one which would present to users a specific web page constructed on demand, aggregating information from various data sources. Using such a system, users would be able to find in one-go the top five information items required for any staff member, aggregated in a single page. We can now calculate both independent and dependent access speed for this hypothetical system. They are $\overline{v_{in}} \approx 4$ and $\overline{v_{de}} \approx 20$ item/min respectively.

Therefore, we can say that the new system would be about ($\frac{4}{3}$ = 1.3) one point three times more efficient in terms of independent access and about ($\frac{20}{4.6}$ = 4.3) four point three times more efficient in terms of dependent access. Not surprisingly, the gain of aggregation and synthesis is significant only on the dependent case, that is when several items of information need to be found.

Knowing people's information access patterns regarding the number of items normally retrieved at one time would allow us to get a more conclusive evaluation of the time gain of our hypothetical system over simply using the intranet. If the answer is mostly one information item, then there is probably not much point to build the envisaged new system since the new system is only marginally more efficient than the current one (1.3 times efficient). However, if what people look for are all information items at once, then there are more incentives (4.3 times efficient) to develop the new system that provides aggregation.

4. Conclusion

One of the advantages for information synthesis and aggregation is that it is claimed to be easier for end users to access relevant information, especially when information comes from multiple heterogeneous data sources. Although that claim is plausible, it has not been verified by any quantitative studies. It is even unclear how one would quantify information access efficiency to be able to compare various approaches to accessing information. In this paper, we define a way to measure information access efficiency which is grounded on information access speed. Information access speed for any information provision system can be determined by following a set of steps.

- identify interaction steps for accessing information items;
- map interaction steps to interaction levels (time);
- calculate information item access time with the access time formula;
- calculate information access speed with the access speed formula.

The average information access speed is then used as criteria for assessing information access efficiency of specific applications in the context of presenting end users with aggregated staff information. This specific study provided us with a simple case where information aggregation and synthesis improves information access efficiency.

To conclude, the significance of our work is three fold. First, to our knowledge, it represents a first attempt to study information access efficiency empirically. It is a step forward in the value proposition of information synthesis and aggregation from qualitative to quantitative measures. Second, our study provided a simple example where it is more efficient to access staff information items when they are aggregated into a single page than accessing them from multiple access points. Third, our study provided an easy and practical approach for gauging whether developing a particular information aggregation application is worthwhile.

Acknowledgements

We wish to thank Ross Wilkinson for originating the ideas of the information access efficiency experiment. We also thank George Ferizis and the anonymous reviewers of the paper for their useful comments. Finally, we thank the people who participated in our survey.

References

[1] Campbell, M.K., DeVellis, B.M., Strecher, V.J. Ammerman, A.S., DeVellis, R.F. and Sandler, R.S. (1994): Improving dietary behavior: The effectiveness of tailored messages in primary care settings. American Journal of Public Health, 84:783–787.

[2] Card, S.K.; Moran, T.P.; and Newell, A. (1983): The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.

[3] Koedinger, K.R.; Anderson, J.R.; Hadley, W.H. & Mark, M.A. (1997): Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8, 30-43.

[4] Newell, A. (1990): Unified theories of cognition. Cambridge, MA: Harvard University Press.

[5] Paris, C., Wu, M., Vander Linden, K., Post, M. and Lu, S. (2004). Myriad: An Architecture for Contextualized Information Retrieval and Delivery, in AH2004: International Conference on Adaptive Hypermedia and Adaptive Web-based Systems. August 23-26, The Netherlands. pp. 205-214.

Comparing XML-IR Query Formation Interfaces

Alan Woodley[1], *Shlomo Geva*[1], *Sylvia Lauretta Edwards*[2]

[1]School of Software Engineering and Data Communications
[2]School of Information Systems
Faculty of Information Technology
Queensland University of Technology
QLD 4000 Australia

{*a.woodley,s.geva,s.edwards*}*@qut.edu.au*

Abstract *XML information retrieval (XML-IR) systems differ from traditional information retrieval systems by using structure of XML documents to retrieve more specific units of information than the documents themselves. Users interact with XML-IR systems via structured queries that express their content and structural requirements. Historically, it has been common belief within the XML-IR community that structured queries will perform better than traditional keyword-only queries. However, recent system-orientated analysis has show that this assumption may be incorrect when system performance is averaged over a set of queries. Here, we test this assumption with users via a simulated work task experiment. We compare a keyword only interface with two user friendly XML-IR interfaces: NLPX, a natural language interface and Bricks, a query-by-template interface. This is the first time that a XML-IR natural language interface has been tested in user experiments. We compare the retrieval performance of all three interfaces and the usability of the two structured interfaces. Our results correspond to those of the system-orientated evaluation and indicate that structured queries do not aid retrieval performance. They also show that in terms of retrieval performance and usability the structured interfaces are comparable.*

Keywords Users, Information Retrieval, XML.

1 Introduction

Traditional information retrieval (IR) systems respond to user queries with a ranked list of relevant documents. XML documents (Figure 1) explicitly separate content and structure. By incorporating structure into the retrieval process XML Information Retrieval (XML-IR) systems are able to return highly specific information to users, lower than the document level. This has the potential to be highly beneficial to users since it can provide very specific responses to their information needs.

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.


```
<article>
    <author>Roger Fuller</author>
    <title>Toward a robust Martian-English
    translator</title>
    <section>
        <title>Introduction</title>
        <paragraph>Because of a dramatic
        lack of interpreters, Communication
        between <bold>Martians</bold> and
        <bold>Terrestrians</bold> is confronted
        to...</paragraph>
    </section>
    <section>
        <title>Introduction</title> ...
    </section>
    ...
</article>
```

Figure 1: XML representation of a scientific article.

The INitiative for the Evaluation of XML Retrieval (INEX) [1] is an organisation that was established to facilitate collaboration between XML-IR researchers. INEX is comparable to TREC [10] and provides a test collection consisting of simulated information needs (topics), a collection of XML documents and for each topic a set of relevant XML elements. INEX has differentiated itself from TREC in two ways: first, by returning results lower than the document level (that is XML Elements) and by facilitating the retrieval of two distinct types of topics - Content Only (CO) and Content and Structure (CAS).

The difference between CO and CAS topics is two fold. First, while they both contain users' content requirements, CAS topics also contain users' structural requirements. For this reason CAS topics are also referred to as structured queries. Second, CO topics express users information needs in keywords while CAS topics express users' information needs in formal languages such as NEXI [6]. A typical CAS topic appears in Figure 2 with its *castitle* element containing a NEXI expression. The addition of structure in CAS topics enables users to write more powerful queries since they are able to direct their search to elements within an XML document that best suit

```
<topic topic_id="275" query_type="CAS" ct_no="131">
    <castitle>//article[about(.//abs, "data mining")]//sec[about(., "frequent itemsets")]</castitle>
    <description>sections about frequent itemsets from articles with abstract about data mining</description>
    <narrative>To be relevant, a component has to be a section about "frequent itemsets". For example, it could be
    about algorithms for finding frequent itemsets, or uses of frequent itemsets to generate rules. Also, the article must
    have an abstract about "data mining". I need this information for a paper that I am writing. It is a survey of different
    algorithms for finding frequent itemsets. The paper will also have a section on why we would want to find frequent
    itemsets.</narrative>
</topic>
```

Figure 2: INEX topic 275 that contains a formal language query (castitle), a natural language query (description) and an information need context (narrative).

their information need. A premise of XML-IR is that this additional information will better fulfil users' information needs, although, recent system based evaluation may contradict this premise [8].

We tested this premise by preforming an interactive XML-IR experiment using both keyword and structured interfaces. For our keyword interface we used a traditional IR interface similar to the interfaces used by Internet search engines. However, a keyword only interface is not able to capture the structural needs of users, which is the reason that INEX has used formal languages such as NEXI [6] to capture users' structural and content requirements. However, formal queries language are too difficult to use by expert - let alone casual users; are too tightly bound to the physical structure of the document and do not scale well across heterogeneous collections. Usability testing has validated the problems that casual users have with formulating NEXI queries [9].

Clearly, a formal language interference is unsuitable for non-laboratory XML-IR use; therefore, XML-IR researchers have investigated alternative user-oriented interfaces. Here, we discuss two such interfaces: NLPX, a natural language interface where users enter queries written in English (such as in Figure 2's *description* element) and Bricks, a query-by-template interface where users enter queries via a graphical user interface. Furthermore, we detail and present outcomes from a useability experiment that compared the retrieval performance of both interfaces with a keyword only interface. This is the first time that a natural language interface has been tested with users. Our results indicate that there is little difference in terms of retrieval between all three interfaces or usability between the structured interfaces.

2 Motivation

The motivation for this research is based on a need to investigate the task of query formation in an interactive XML-IR setting, a gap in current knowledge. Our research focus manifests itself in two areas. First, we want to observe if the addition of structural requirements to queries aids in retrieval. Second, we want to investigate alternative means of users formulating structured queries.

2.1 Previous XML-IR Evaluation

The majority of XML-IR system evaluation has been batch testing using a version of the Cranfield Methodology, which uses a controlled set of queries and relevance judgments [3]. Historically, this method of evaluation has been very successful for evaluating and improving the retrieval performance of IR systems and algorithms, particularly in traditional IR [10]. This method allows for repeated and extensive testing of systems within a laboratory setting, however, since the method does not involve actual users there is no way of guaranteeing that their full needs are being met. In fact, research on traditional IR systems has shown that improved retrieval performance in a laboratory environment does not always correlate with user satisfaction [5]. The field of interactive information retrieval evaluation was established to collect quantitative and qualitative feedback from users regarding their use of IR systems. However, most interactive XML-IR experiments [7] have focused on results presentation (for instance: do users prefer to read several paragraphs or one section) rather than query formation. This work presents one of the first investigations into how users formulate queries for use in an XML-IR system. And in particular, the first time that a natural language interface has been compared to other interfaces,

2.2 Adding Structural Hints to Queries

XML-IR system return document fragments (elements) rather than entire documents. A common belief amongst XML-IR researchers has been that adding structural hints to queries will improve retrieval performance, in particular precision. The premise stems from a belief that by adding structural hints users will be able to focus retrieval more closely to elements that match their information need. Historically, this premises has not been verified. Even though INEX has had sperate tracks that deal specifically with content only queries (CO) and content and structure queries (CAS), they have always used different topics; thereby, disabling valid comparison between the two tracks.

In 2005, INEX decided to verify this premise by the introduction of an additional CO+S track [2]. The premises of the CO+S track was for a user to perform

```
<topic topic_id="202" query_type="CO+S" ct_no="1">
    <title>ontologies case study</title>
    <castitle>//article[about(., ontologies)]//sec[about(., ontologies case study)]</castitle>
    <description>Case studies in the use of ontologies</description>
    <narrative>I'm writing a report on the use of ontologies. I'm interested in knowing how ontologies are used to
    encode knowledge in real world scenarios. I'm particularly interested in knowing what sort or concepts and relations
    people use in their ontologies. I'm not interested in general ontology frameworks or technical details about tools for
    ontology creation or management. An example relevant result contains a description of the real world phenomena
    described by the ontology and also lists some of the concepts used and relations between concepts. </narrative>
</topic>
```

Figure 3: INEX topic 202 that contains both a CO query (title) and a CO+S query (castitle)).

a standard IR interaction using an initial content only query, however, if the user was unsatisfied with the results list (for instance it contained too many irrelevant elements) then he/she could narrow down their search by creating a second version that contained a structural hint. CO and CO+S queries were encapsulated within the same topic, and shared the same set of relevance judgements thereby allowing for valid comparison. Figure 3 is an example of a CO/CO+S topic. The CO query is expressed as keywords in the title element, and the CAS query is expressed in formal language in the castitle element.

The retrieval performance of participants in both the CO and CO+S tracks was analysed by Trotman and Lalmas [8]. They showed that while some systems performed better in the CO+S track, none of the improvements were statistically significant. They concluded that the reason that the addition of structure did not improve retrieval performance was because users were not able to write meaningful structured queries. However, they also suggest that this may be a problem of the INEX's source collection. A third alternative, not suggested by the authors, was that current XML-IR systems are not able to process structured queries effectively.

Regardless of the outcomes of Trotman and Lalmas' study, we feel that it is important to observe the effect of adding structure to queries within the field of XML-IR, for several reasons. First, while the addition of structure to XML-IR queries has been investigated in system-orientated (or Cranfield-like) testing, it has not been fully investigated in interactive XML-IR experiments. Secondly, the addition of structure may help retrieval in different collections or if it is better handled by XML-IR systems. It should also be noted that the IR system used as in our experiments performed better using CO+S queries, rather than CO, in batch testing.

2.3 XML-IR Interfaces

There are two standard interfaces for interacting with XML-IR systems, keywords and structured formal languages; however, neither interface optimally addresses the needs of XML-IR users. Here, we discuss the problems associated with both types of interfaces, and outline how they can be solved using alternative interfaces such as natural language or query-by-template. First, keyword based interfaces are too unsophisticated to fully capture XML-IR users' complex information needs since users are unable to specify structural constraints. For instance, in the information need present in Figure 2 the user only wants to search in abstracts and sections, which they are unable to specify just using keywords. Secondly, users may wish to search parts of documents that they do not intend to retrieve, but are rather used to aid (or support) their retrieval. For example, in the information need present in Figure 2 the user wants to retrieve sections from articles that with an abstract on data mining, however, they do not wish to retrieve the abstract itself. Again, this information can not be conveyed just using keywords.

The complexity of XML-IR has lead to the development of formal query languages (akin to SQL for databases) specifically designed for XML-IR, such as NEXI [6]. A sample NEXI expression is presented in the castitle tag in Figure 2. However, formal query languages have also posed problems. First, formal query languages are too difficult for users, both expert and casual, to correctly express their structural and content information needs. Examples of difficulties experienced by expert users occurred at the 2003 and 2004 INEX Workshops where 63 per cent and 12 per cent of queries constructed by experts had major semantic or syntactic errors. It has already been shown that XML-IR users find query-by-template interfaces easier to use than formal language interfaces [9] and users should be able to intuitively express their information need in a natural language. Second, formal query languages are too tightly bound to the physical structure of documents; hence, users need to know the physical tag names of elements in order to express their structural needs. While this information may be obtained from a document's DTD or Schema, users are unlikely to remember hundreds of tags names; furthermore, due to security/privacy reasons, there are situations where the proprietor of the collection does not wish to grant public access to those files. The problem is magnified in heterogeneous collections since a single tag can have multiple names. In contrast, structural requirements in both natural language and

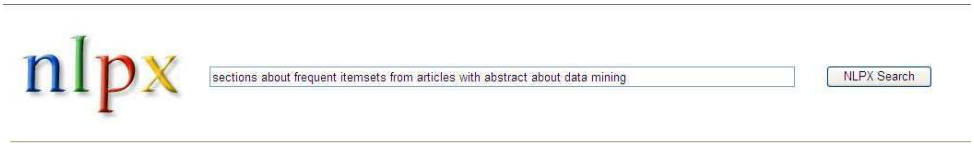


Figure 4: NLPX input using the information need in Figure 2 as a sample query

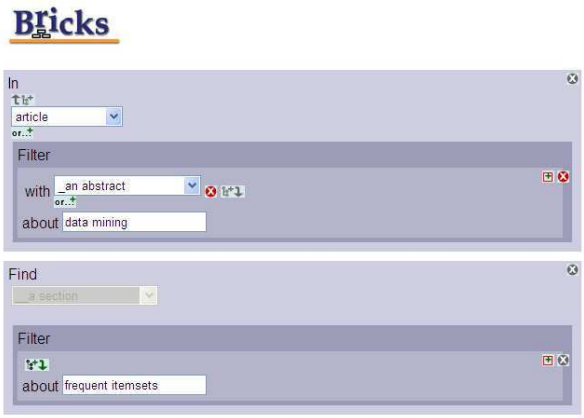


Figure 5: Bricks input using the information need in Figure 2 as a sample query

query-by-templates can be expressed at a higher conceptual level, allowing the underlying documents' structure to be completely hidden from users

3 Experimental System

The system used in the experiment is separated into two parts: the front-end interfaces and the backend retrieval system. Two different interfaces were used: NLPX, that accepted queries written in natural language (English) [11][12], and Bricks, a query by template interface that allowed users to enter queries via a graphical user interface [9]. Examples of the input screen used for both interfaces appear in Figure 4 and Figure 5. These examples capture the information need expressed in the description element of Figure 2 and are a representative of the type of queries entered by the participants. The same backend search engine, GPX, was used for both interfaces. Since GPX only accepted formal language queries, both interfaces translated their user input into NEXI before submitting them to GPX. Below we describe NLPX, Bricks and GPX in more detail.

3.1 Interface A - NLPX

NLPX accepts natural language queries (NLQs) and produces formal queries written in the NEXI language. The NLPX translation process involves four steps. First, NLPX tags words either as special connotations (for instance structures) or by their part of speech. Second NLPX divides sentences into atomic, non overlapping segments (called chunks) and then classifies them into grammatical classes. Third, NLPX matches the tagged NLQs to query templates that were derived from the inspection of previous INEX queries. Finally, NLPX outputs the query in NEXI format. Batch testing of a single backend search engine that used both natural language queries parsed through NLPX and formal NEXI queries has shown comparable results [12]. This is the first time that NLPX has been tested in a usability experiment.

3.2 Interface B - Bricks

Bricks is a query-by-template interface that allows users to input structured queries via a graphical user interface (GUI). Users enter their content needs via text boxes and their structural needs via drop-down boxes. To aid users, structural needs were indicated via conceptual rather than physical names, for example "a section" rather than sec. Bricks allows users to develop queries in several steps ("blocks") starting with their desired unit of retrieval and then by adding any additional information needs. Blocks were also added as the user traversed the hierarchy of the documents (for instance from *article* to *section* to *paragraph*). Upon completion of input, the data in the Bricks GUI was translated to formal NEXI expression, however, due to the constraints of the GUI, users were unable to enter malformed expressions. Usability testing has shown that users find Bricks superior to keyword only and NEXI interfaces [9].

3.3 Backend Retrieval System- GPX

The backend retrieval system for this experiment was Gardens Point X (GPX) [4]. GPX was chosen since it has performed strongly at the annual INEX conference since 2002 - consistently among the top three systems. GPX stores the information about each leaf element in the collection as an inverted list. Upon retrieval, GPX

Table 1: The order of the information needs us by each user group

Sub-Experiment (interface)	1 (NLPX)		2 (NLPX)		3 (Bricks)	
Topic Order	1	2	3	4	5	6
Group A	253	256	257	270	275	284
Group B	275	284	253	256	257	270
Group C	257	270	275	284	253	256

matches query terms to all leaf elements that contain the term and then dynamically creates their ancestors. Elements are ranked according to their predicted relevance in GPX's ranking scheme. GPX rewards leaf elements that contain phrases and specific, rather than common, terms. It also rewards ancestors with multiple relevant children, rather than a single relevant child. For this experiment, the results list was filtered so that "overlapping elements" (that is, elements whose ancestors or descendants appear higher ranked on the results list) were removed before been being presented to users. This decision was made because users have been known to react negatively to overlapping elements [7].

4 Experimental Methodology

4.1 Participants, Collection and Information Requests

The experiment simulated the task of users interacting with an academic retrieval system. Sixteen participants took part in the experiment. The participants acted as academic researchers, for example: post-graduate research students, corporate researchers or academics. The participants searched a collection of academic journal articles, specifically IEEE journal articles from 1995 to 2002. The journals had a broad range of focus, ranging from general journals such as Computing to specific journals such as Neutral Networks (the complete list can be foundd in the annual INEX proceedings [1] [2]).

The participants were post-graduate information technology students who were uninitiated in the domain of XML-IR. While this may not be a representative sample of possible XML-IR users, it was necessary to have such participants since understanding the technical nature of the information needs and source collection was beyond casual users. Also since the participants were uninitiated in the domain of XML-IR, it is valid for us for us to extrapolate the results of this experiment into the wider area of XML-IR.

The participants were given six information needs that simulated those of a real user. The information needs contained both a detailed explanation of the information sought and a condition of relevance that described the motivation behind the information need. The information needs were sampled from the narrative elements of INEX Topics 253 - 284; an example information need was presented in the narrative element of Figure 2.

4.2 Sub-Experiments

The participants preformed three sub-experiments that correlated to three different methods of translating information needs. The first two sub-experiments used the NLPX interface, whereas the last sub-experiment used the Bricks interface. For each sub-experiment, the participants attempted to fulfil two information needs. To reduce bias, the participants where split into three groups and used the information needs in the order presented in Table 1. For each information need the participants interacted with the interfaces by submitting queries and receiving back matching information items, which may or may not be relevant.

The first sub-experiment examined users' initial reaction to using the NLPX interface. They were instructed to enter keyword only queries into NLPX as if it were a standard Internet search engine. Participants were then given a short tutorial about structured information retrieval and were shown some examples of structured natural language queries. The participants then performed the second sub-experiment by entering structured queries into NLPX. A second tutorial was then given on how to use the Bricks interface to preform structured queries. Following this, the participants performed the final sub-experiment were they entered structured queries using the Bricks interface.

Following the experiment, feedback from participants was sought in two ways: first, a survey conducted directly after the experiment and second, one-on-one interviews conducted in the weeks following the experiment that were recorded and later transcribed. During the experiment, the actions of the participants, such as: the queries they entered, the information items they viewed, and their relevance judgments, were logged to allow for quantitative analysis. Participants' confidentiality was maintained throughout the experiment. Before the experiment began, participants were made aware of all feedback sought and were given the option of not participating in the experiment, however, all decided to participate. The participants signed a permission form to ensure that the feedback results could be published. Furthermore, clearance was sought and approved by QUT's ethics committee.

Table 2: The number of relevant elements retrieved by each interface.

	TOPIC NUMBER						
	253	256	257	270	275	284	Average
KO	6	27	64	8	55	45	34.2
NLPX	7	33	22	11	82	5	26.7
Bricks	1	12	22	11	48	5	16.5

Table 3: The ratio of relevant elements retrieved to total number of elements retrieved by each interface.

INTERFACE	TOPIC NUMBER						
	253	256	257	270	275	284	Average
KO	0.084	0.458	0.293	0.097	0.467	0.281	0.280
NLPX	0.069	0.620	0.197	0.162	0.660	0.095	0.300
BRICKS	0.018	0.267	0.215	0.141	0.863	0.192	0.283

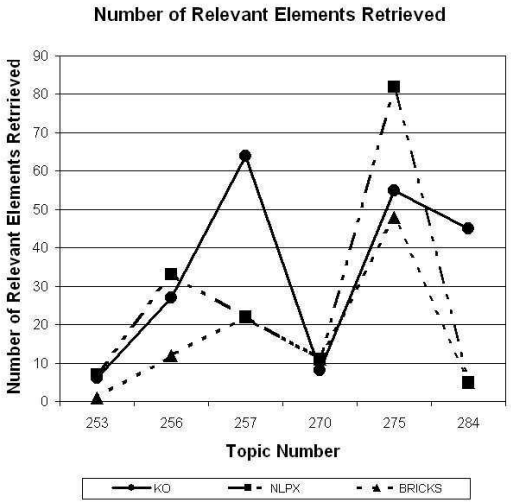


Figure 6: The number of relevant elements retrieved by each interface.

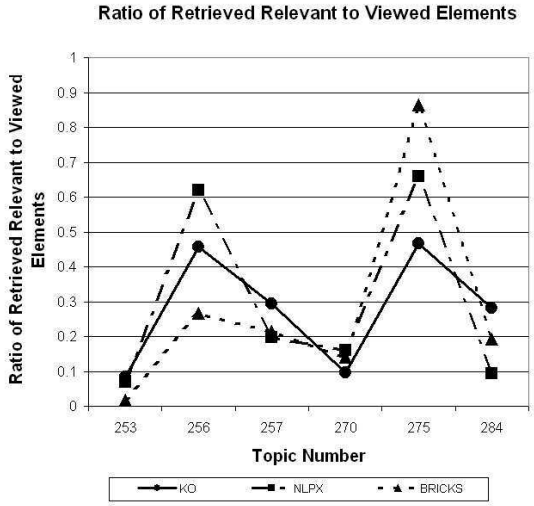


Figure 7: The ratio of relevant elements retrieved to total number of elements retrieved by each interface.

5 Results

5.1 Retrieval Performance

Here, we present the results from our experiments. Our investigation consists of a mixture of quantitative and qualitative analysis. Hence. we present two sets of results, first, the retrieval performance of the three interfaces and second, the results of a survey that examined the usability of both NLPX and Bricks. Official INEX relevance judgements were used in our analysis, thereby, keeping the relevance judgements consistent across participants, since we wanted to narrow the scope of our research to the performance of interfaces.

Table 2 and Figure 6 present the number of relevant elements retrieved by each of the interfaces for the six INEX topics. This is a recall orientated measure. As the results show, there is no significant difference between keywords only and the structured interfaces (NLPX and Bricks), in fact on average the structured interfaces perform worse than keywords only. This finding corresponds to the work of Trotman and Lalmas [8]. Table 3 and Figure 7 present the average ratio of relevant results retrieved to those viewed by users. This is a precision orientated measure. Here, the results for NLPX outperform both the keywords only and Bricks interfaces, however, once again the results are not significant.

5.2 Usability Scores

The second area we investigated was how the two different interfaces, NLPX and Bricks, affected the retrieval experience of the users. Following the experiment we asked the participants five questions about their experience using each interface. For each interface, participants were asked to respond to each question with a rating between 1 and 10 on how well the interface successfully fulfilled a set criterion for that interface. The questions and the average response for all participants presented in Figure 4 and Table 8. The results indicate that the participants did not think that there was much variation between the two interfaces. In particular, the averages for questions two, three and four were almost identical between interfaces. There was a slight difference between interfaces in the averages for questions one and five, however, neither difference is statistically significant.

Table 4: Participants survey results regarding interface usability.

	NLPX	Bricks
Easy to Use	5.313	4.563
Found Relevant Results	4.938	5.000
Ranked Results Highly	4.813	4.750
Accurately expressed my information need	4.500	4.563
Fully expressed my information need	4.00	4.38

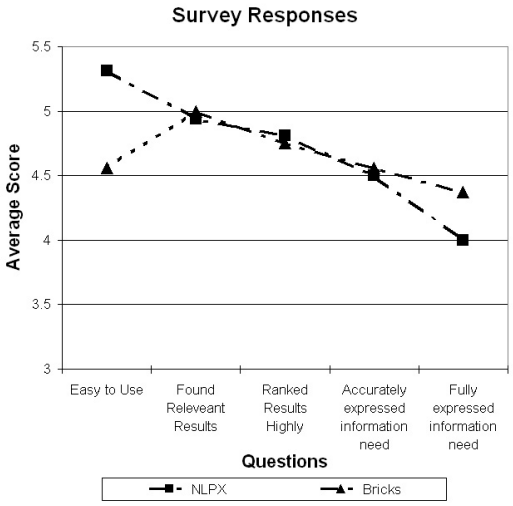


Figure 8: Participants survey results regarding interface usability.

6 Discussion

This research was motivated by a desire to investigate how users interact with XML-IR systems, in particular, how users formulate XML-IR queries. Our experiments focused on the retrieval performance of keyword only and structured query interfaces, and the usability of structured interfaces.

Prior to the study of Trotman and Lalmas [8] it was believed that structured queries would outperform traditional keyword only queries in XML-IR. However, their research disproved this assumption. Our findings correspond to those of Trotman and Lalmas. In fact, there was only one instance were the retrieval performance of NLPX conclusively outperformed that of the keyword only systems. This was for INEX topic 275 presented in Figure 2. Note, that the information need (narrative) for this topic is very direct, specifically asking for sections and abstracts containing certain content items. This may be a reason why it performed so well in our experiment.

We do not know why the keywords only interface performs as strongly as the structured interfaces. It may be, as suggested by Trotman and Lalmas, that users are unable to formulate effective structured queries - that is they are unable to identify which structures in the collection contain relevant information. This possibility is especially pertinent to this experiment since since the participates were uninitiated in the domain of XML retrieval and would therefore be less likely to write effective structured queries than experts. Alternatively it may be as a consequence of INEX IEEE collection since most of the retrieved elements are syntactic (for instance section, paragraph) rather than semantic in nature.

However, we also showed that for structured retrieval, a natural language interface is as effective as a query by template interface. This is important contribution, seeing that this is the first time that a natural language interface has been tested in a usability experiment. According to the user surveys the only difference between the two interfaces was in ease of use and fullness of capturing information need. Users felt that NLPX was easier to use than Bricks, which could show that natural language interfaces are more intuitive than query-by-template interfaces. In contrast, users felt that Bricks captured their information need more fully than NLPX, possibly due to that fact that users were unsure to the degree that NLPX was correctly interpreting their queries.

7 Future Work

Since this was a pilot study on users' interaction with XML-IR system there remains much to be investigated. Here, we outline further research that could be conducted based upon our pilot study.

More participants. Our experiments contained sixteen participants which is not statistically significant for quantitative analysis. However, a larger number of participants (for example fifty to a hundred) would provide a statistically significant number for quantitative analysis while strengthening the qualitative testimony.

Wider pool of participants. The participants in our experiment were post-graduate information technology students which is not representative of the types of users that could possibly use XML-IR systems. In our experiment this was a necessary constraint since the source collection was restricted to IEEE journals. However, if a more general collection was used, then a wider range of participants could be used.

More guidance on how to use NLPX. Participants in our experiment were given minimal guidance on how to use NLPX. This was by design, since we wanted to observe how uninitiated users would

interact with a natural query interface. However, some participants found this disconcerting. It would be interesting to see how further guidance would effect users retrieval experience.

Alternative collection As stated, an alternative source collection would allow for a wider pool of participants. Another justification for an alternative collection is that the IEEE collection was syntactic rather than semantic in nature. Hence, it may not really matter to users if a certain term appears in a paragraph or an abstract. Whereas, if the source collection contained information about movies that it could be very important if the movie was *titled Capote* or if it was *written by Truman Capote*.

Longer time span. Our experiment was conducted within a two hour period. If a similar experiment was conducted over a longer period, for example twelve eighteen months, then more data could be collected and analysed. For instance we could observe if the users interaction with the interfaces changed over time. This would provide valuable information on real users interaction with XML-IR systems.

8 Conclusion

We observed how uninitiated users interact with XML-IR systems and recorded their experience. Our results do not show that incorporating structural hints into queries aids retrieval. However, when structural hints are added to queries our results indicate that users experience with a natural language interface is similar to their experience with a query-bytemplate interface, the current standard for interactive XML-IR systems. These results indicate that further research in is warranted to attain further understanding of XML-IR users needs.

Acknowledgements The authors would like to acknowledge the anonymous participants of the experiments, in particular those participants who were interviewed following the experiment. Without their contribution this work could not been conducted.

References

[1] Norbert Fuhr, Norbert Gövert, Gabriella Kazai and Mounia Lalmas (editors). *INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the First INEX Workshop. Dagstuhl, Germany, December 8–11, 2002*, ERCIM Workshop Proceedings, Sophia Antipolis, France, March 2003. ERCIM. `http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf`.

[2] Norbert Fuhr, Mounia Lalmas, Saadia Malik and Gabriella Kazai (editors). *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*, Volume 3977 of *Lecture Notes in Computer Science*. Springer, 2006.

[3] Norbert Fuhr, Mounia Lalmas, Saadia Malik and Zoltán Szlávik (editors). *Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6-8, 2004, Revised Selected Papers*, Volume 3493 of *Lecture Notes in Computer Science*. Springer, 2005.

[4] Shlomo Geva. GPX - Gardens Point XML-IR at INEX 2005. In *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*.

[5] William R. Hersh, Andrew Turpin, Susan Price, Dale Kraemer, Benjamin Chan, Lynetta Sacherek and Daniel Olson. Do batch and user evaluations give the same results? an analysis from the trec-8 interactive track. In *TREC*, 1999.

[6] Richard A. O'Keefe and Andrew Trotman. The simplest query language that could possibly work. In *The Proceedings of the 2003 INEX Workshop*, 2004.

[7] Jovan Pehcevski, James A. Thom, Seyed M. M. Tahaghoghi and Anne-Marie Vercoustre. Hybrid xml retrieval revisited. In Fuhr et al. [2], pages 153–167.

[8] Andrew Trotman and Mounia Lalmas. Why structural hints in queries do not help xml-retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 711–712, New York, NY, USA, 2006. ACM Press.

[9] R. Van Zwol, J. Baas, H. Van Oostendorp and F. Wiering. Query formulation for xml retrieval with bricks. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, 2005.

[10] Ellen M. Voorhess and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, Massachusetts, 2005.

[11] Alan Woodley and Shlomo Geva. Nlpx at inex 2004. In *Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6-8, 2004, Revised Selected Papers*, pages 382–394, 2005.

[12] Alan Woodley and Shlomo Geva. Nlpx at inex 2005. In *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*, pages 358–372, 2006.