Available online at www.sciencedirect.com
ScienceDirect
Information Processing and Management 43 (2007) 169–180
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Using query logs to establish vocabularies in distributed information retrieval

Milad Shokouhi *, Justin Zobel, Saied Tahaghoghi, Falk Scholer

School of Computer Science and Information Technology, RMIT University, Melbourne 3001, Australia

Received 11 December 2005; accepted 3 April 2006
Available online 7 July 2006

Abstract

Users of search engines express their needs as queries, typically consisting of a small number of terms. The resulting search engine query logs are valuable resources that can be used to predict how people interact with the search system. In this paper, we introduce two novel applications of query logs, in the context of distributed information retrieval. First, we use query log terms to guide sampling from uncooperative distributed collections. We show that while our sampling strategy is at least as efficient as current methods, it consistently performs better. Second, we propose and evaluate a pruning strategy that uses query log information to eliminate terms. Our experiments show that our proposed pruning method maintains the accuracy achieved by complete indexes, while decreasing the index size by up to 60%. While such pruning may not always be desirable in practice, it provides a useful benchmark against which other pruning strategies can be measured.

Keywords: Distributed information retrieval; Uncooperative environments; Indexing; Query logs

1. Introduction

Traditional information retrieval systems use corpus, document, and query statistics to identify likely answers to users' queries. However, these queries can be captured in a query log, providing an additional source of evidence of relevance. In recent years, considerable attention has been devoted to the study of query logs and the way people express their information needs (de Moura et al., 2005; Fagni, Perego, Silvestri, & Orlando, 2006; Jansen & Spink, 2005). The query logs of commercial search engines such as Excite[1] (Spink, Wolfram, Jansen, & Saracevic, 2001), Altavista[2] (Silverstein, Marais, Henzinger, & Moricz, 1999), and AlltheWeb[3] (Jansen & Spink, 2005) have been investigated and analysed. Query logs have been used in information

* Corresponding author.
E-mail address: milad@cs.rmit.edu.au (M. Shokouhi).
[1] http://www.excite.com.
[2] http://www.altavista.com.
[3] http://www.alltheweb.com.

0306-4573/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2006.04.003

retrieval research for applications such as query expansion (Billerbeck, Scholer, Williams, & Zobel, 2003; Cui, Wen, Nie, & Ma, 2002), contextual text retrieval (Wen, Lao, & Ma, 2004), and image retrieval (Hoi & Lyu, 2004). The question we explore in this paper is how query logs can be used to guide future search, in the context of distributed information retrieval.

In distributed information retrieval (DIR) systems, the task is to search a group of separate collections and identify the most likely answers from a subset of these. *Brokers* receive queries from the users and send them to those collections that are deemed most likely to contain relevant answers. In a *cooperative* environment, collections inform brokers about the information they contain by providing information such as term distribution statistics. In *uncooperative* environments, on the other hand, collections do not provide any information about their content to brokers. A technique that can be used to obtain information about collections in such environments is to send *probe queries* to each collection. Information gathered from the limited number of answer documents that a collection provides in response to such queries is used to construct a *representation set*; this representation set guides the evaluation of user queries.

In this paper, we introduce two novel applications of query logs: sampling for improved query probing, and pruning of index information.

The first of these is sampling. Using a TREC web crawl, we show that query log terms can be used produce effective samples from uncooperative collections. We compare the performance of our strategy with the state-of-art method, and show that samples obtained using query log terms allow for more effective collection selection and retrieval performance – improvements in average precision are often over 50%. Our method is at least as efficient as current sampling methods, and can be much more efficient for some collections.

Our second new use of query logs is a pruning strategy that uses query log terms to remove less significant terms from collection representation sets. For a DIR environment with a large number of collections, the total size of collection representation sets on the broker might become impractically large. The goal of pruning methods is to eliminate unimportant terms from the index without harming retrieval performance. In previous work – such as that of Carmel et al. (2001), Craswell, Hawking, and Thistlewaite (1999), de Moura et al. (2005) and Lu and Callan (2002) – pruning strategies have had an adverse effect on performance. The reason is that these approaches drop many terms that are necessary to future queries. We show that pruning based on query logs does not decrease search precision. In addition, our method can be applied during document indexing, which means that it independent of term frequency statistics. We also test our method on central indexes and for different types of search tasks. We show that, by applying our pruning strategy, the same performance as a full index can be achieved, while substantially reducing index size. In practice, such pruning might not always be desirable; if a term is present, it should be searchable. However, our pruning does provide an interesting benchmark against which other methods can be measured, and is clearly superior to the principal alternative.

2. Distributed search

The vast volume of data on the web makes it extremely costly for a single search engine to provide comprehensive coverage. Moreover, public search engines cannot crawl and index documents to which there are no public links, or from which crawlers are forbidden. These documents form the so-called *hidden web* and generally only be viewed by using custom search interfaces supplied as part of the site. Distributed information retrieval (DIR) aims to address this issue by passing queries to multiple servers through a central broker. Each server sends its top-ranked answers back to the broker, which produces a single ranked list of answers for presentation to the user. For efficiency, the broker usually passes the query only to a subset of available servers, selecting those that are most likely to contain relevant answers. To identify the appropriate servers, the broker calculates a similarity between the query and the representation set of each server.

In cooperative environments, servers provide the broker with their representation sets (Callan, Lu, & Croft, 1995; Fuhr, 1999; Gravano, Chang, Garcia-Molina, & Paepcke, 1997, 1999; Yuwono & Lee, 1997). The broker can be aware of the distribution of terms at the servers, and is therefore able to calculate weights for each server. Queries are sent to those servers that indicate the highest weight for the query terms.

In practice, servers may be uncooperative and therefore do not publish their index information. Server representation sets can be gathered using *query-based sampling* (QBS) (Callan, Connell, & Du, 1999). In QBS, an

initial query is created from frequently-occurring terms found in a reference collection – to increase the chance of receiving an answer – and sent to the server. The query results provided by the server are downloaded, and another query is created using randomly-selected terms from these results. This process continues until a sufficient number of documents have been downloaded (Callan & Connell, 2001; Callan et al., 1999; Shokouhi, Scholer, & Zobel, 2006). Many queries do not return any yet-unseen answers; Ipeirotis and Gravano (2002) claim that, on average, one new document is received per two queries. QBS also downloads many pages that are not highly representative for the server.

2.1. Query-based sampling

Query-based sampling QBS was introduced by Callan et al. (1999), who suggested that even a small number of documents (such as 300) obtained by random sampling can effectively represent the collection held at a server. They tested their method on the CACM collection (Jones & Rijsbergen, 1976) and many other small servers artificially created from TREC newswire data (Voorhees & Harman, 2000). In QBS, subsequent queries after the first are selected by choosing terms from documents that have been downloaded so far (Callan et al., 1999). Various methods were explored; random selection of query terms was found to be the most effective way of choosing probe queries, and this method has since been used in other work on sampling non-cooperative servers (Craswell, Bailey, & Hawking, 2000; Si & Callan, 2003). These methods generally proceed until a fixed number of documents (usually 300) have been downloaded. However, Shokouhi et al. (2006) have shown that for more realistic, larger collections, fixed-size samples might not be suitable, as the coverage of the vocabulary of the server is poor.

An alternative technique, called *Qprober* (Gravano, Ipeirotis, & Sahami, 2003), has been proposed for automatic classification of servers. Here, a classification system is trained with a set of pages and judgments. Then the system suggests the classification rules and uses the rules as queries. For example, if the classification system suggests (*Madonna* → *Music*), it uses "Madonna" as a query and classifies the downloaded pages as music-related. Qprober differs from QBS in the way that probe queries are selected and requires a classification system in the background.

2.2. Using query logs for sampling

Terms that appear in search engine query logs are – by definition – popular in queries, and tend to refer to topics that are well-represented in the collection. We therefore hypothesis that probe queries composed of query log terms would return more answers than the random terms, leading to higher efficiency. Since query terms are aligned with actual user interests, we also believe that sampling using query log terms would better reflect user needs than random terms from downloaded documents. Hence, instead of choosing the terms from downloaded documents for probe queries, we use terms from query logs. Analysis of our method shows that it is at least as efficient as previous methods, and generates samples that produce higher overall effectiveness.

2.3. Evaluation

To simulate a DIR environment, we extracted documents from the 100 largest servers in the TREC WT10 g collection (Bailey, Craswell, & Hawking, 2003). These sets vary in size from 26 505 documents (www9.yahoo.com), to 2790 documents (swax32.-swmed.edu), with an average size of 5602 documents per server. For sampling queries, we used the 1000 most frequent terms in the Excite search engine query logs collected in 1997 (Spink et al., 2001).

For each query, we download the top 10 answers; this is the number of results that most search interfaces return on the first page of results. Sampling stops after 300 unique documents have been downloaded or 1000 queries have been sent (whichever comes first). Although using fixed-size samples might not always be the optimal method (Shokouhi et al., 2006), we restrict ourselves to 300 documents to ensure that our results are comparable to the widely accepted baseline (Callan et al., 1999).

For each server we gather two samples: one by query-based sampling, and the other by our query log method. For query log (QL) experiments, each of the 1000 most frequent terms in the Excite query logs are

Table 1
Comparison of the QL and QBS methods on a subset of the WT10 g data; QL consistently performs better

CO	MAP		P@5		P@10		*R*-precision	
	QBS	QL	QBS	QL	QBS	QL	QBS	QL
1	0.0668	0.0902	0.1302	0.1721	0.0744	0.0988	0.0744	0.0988
10	0.1562	0.2515‡	0.3057	0.4322‡	0.2011	0.3023‡	0.2011	0.3023‡
20	0.1617	0.2811‡	0.3149	0.4621‡	0.2115	0.3437‡	0.2115	0.3437‡
30	0.1540	0.2655‡	0.2941	0.4471‡	0.2106	0.3259‡	0.2106	0.3259‡
40	0.1812	0.2639‡	0.3200	0.4306‡	0.2459	0.3212‡	0.2459	0.3212‡
50	0.1868	0.4188‡	0.3341	0.4188†	0.2506	0.3176‡	0.2506	0.3176‡

Differences that are statistically significant based on the *t*-test at the 0.05 and 0.01 level of significance are indicted by † and ‡, respectively. "CO" is the cutoff number of servers from which answers are retrieved.

passed as a probe query to the collection, and the top 10 returned answers are collected. For QBS, probe queries are selected from the current downloaded documents at each time, and the top 10 results of each query are gathered.

To evaluate the effectiveness of samples for different queries, we used topics 451–550 from the TREC-9 and TREC 2001 Web Tracks. We used only terms in the `title` field as queries. Since we are extracting only the largest 100 servers from WT10 g, the number of available relevant documents is low, so the precision-recall metrics produce poor results. For this reason, many DIR experiments use the set of documents that are retrieved by a central server as an oracle. That is, all of the top-ranked pages returned by the central index are considered to be relevant, and the performance of DIR approaches is evaluated based on how effectively they can retrieve this set (Craswell et al., 2000; Xu & Callan, 1998). Therefore, we use a central index containing the documents of all 100 servers[4] as a benchmark.

For both the baseline and DIR experiments, we gathered the top 10 results for each query. Results for 100 and 1000 answers per query were found to be similar and are not presented here. We tested different cutoff (CO) points in our evaluations: for a cutoff of 1, the queries were passed to the one server with the most similar corresponding representation set; for a cutoff of 50, queries were sent to the top 50 servers. Table 1 shows that the QL method consistently produces better results. Differences that are statistically significant based on the *t*-test at the 0.05 and 0.01 level of significance are indicted by the † and ‡, respectively. For mean average precision (MAP), which is considered to be the most reliable evaluation metric (Sanderson & Zobel, 2005), QL outperforms QBS significantly in four of five cases.

We made two key observations. First, query log (QL) terms did not retrieve the expected 300 documents for four servers after 1000 queries, while QBS failed to retrieve this number from only one server. Analysis showed that these servers contain documents unlikely to be of general interest to users. For example www.twobirds.com has error pages and HTML forms while www.snweb.com includes many pages with non-text characters.

Second, the QL method downloads an average of 2.43 unseen documents per query, while the corresponding average for QBS is 2.80.

Having access to the term document frequency information of any collection, it is possible to calculate the expected number of answers from the collection, for single-term queries extracted randomly from its index. Therefore, we indexed all of the servers together as a global collection. At most 10 answers are retrieved per query. The expected number of answers per query can be calculated as

$$\frac{|\text{Number of Terms df} > 9|}{|\text{Total Number of Terms}|} \times 10 + \sum_{i=1}^{9} \frac{|\text{Number of Terms df} = i|}{|\text{Total Number of Terms}|} \times i$$

which gives an expected value of 2.60, close to numbers obtained by both the QL and QBS methods.

[4] The 100 servers consist of 563 656 documents in total, containing 309 195 668 terms, 1 788 711 of them unique.

Table 2
Comparison of the QL and QBS methods, showing average number of answers returned per query

Collection	Size	Unseen (QBS)	Total (QBS)	Unseen (QL)	Total (QL)
Newswire	30507	4.6	5.8	4.9	9.1
WEB-1	304035	1.8	2.2	6.9	9.9
WEB-2	218489	2.5	2.9	6.6	9.9
WEB-3	817025	1.2	1.5	7.4	9.9
GOV-1	136176	2.5	3.8	7.3	9.7

However, these values contrast with those reported by Ipeirotis and Gravano (2002), who claim that QBS downloads an average of only one unseen document per two queries. On further investigation, we observed that the average varies for different collections, as shown in Table 2. The first collection is extracted from TREC AP newswire data and contains newspaper articles. Collections labelled WEB are subsets of the TREC WT10 g collection (Bailey et al., 2003). Finally, GOV-1 is a subset of the TREC GOV1 collection (Craswell & Hawking, 2002). Note that the average values for QL are between 4.9 and 7.4 unseen documents per query, while for QBS these range from 1.2 to 4.5. In general, the gap between methods is more significant for larger collections with broad topics. Each QL probe query returns about 10 answers – the maximum – on the first page, while this number is considerably lower for QBS.

3. Pruning using query logs

In uncooperative DIR systems, the broker keeps a representation sample for each collection (Callan & Connell, 2001; Craswell et al., 2000; Shokouhi et al., 2006). These samples usually contain a small number of documents downloaded by query-based sampling (Callan et al., 1999) from the corresponding collections.

Pruning is the process of excluding unimportant terms (or unimportant term occurrences) from the index to reduce storage cost and, by making better use of memory, increase querying speed. The importance of a term can be calculated according to factors such as lexicon statistics (Carmel et al., 2001) or position in the document (Craswell et al., 1999). The major drawback with current pruning strategies is that they decrease precision, because the pruned index can miss terms that occur in user queries. In addition, in *lexicon-based* pruning strategies, the indexing process is slowed significantly. First, documents need to be parsed, so that term distribution statistics are available. Then, unimportant terms can be identified, and excluded, based on the lexicon statistics. For example, terms that occur in a large proportion of documents might be treated as stopwords. Finally, the index needs to be updated based on the new pruned vocabulary statistics.

Lexicon-based pruning strategies face additional problems when dealing with broker indexes in DIR. Documents are gathered from different collections, with different vocabulary statistics. A term that appears unimportant in one collection based on its occurrences might in fact be critical in another collection. Therefore, pruning the broker's index based on the global lexicon statistics does not seem reasonable.

We introduce a new pruning method that addresses these problems. Our method prunes during parsing, and is therefore faster than lexicon-based methods, as index updates are not required. Unlike other approaches, our proposed method does not harm precision, and can increase retrieval performance in some cases. Note, however, that we regard this pruning strategy as an illustration of the power of query logs rather than a method that should be deployed in practice: users for search for a term should be able to find matches if they are present in the collection. Although sampling inevitably involves some loss, that loss should be minimised. That said, as our experiments show the new pruning method is both effective and efficient.

3.1. Related work

Pruning is widely used for efficiency, either to increase query processing speed (Persin, Zobel, & Sacks-Davis, 1996), or to save disk storage space (Carmel et al., 2001; Craswell et al., 1999; de Moura et al., 2005; Lu & Callan, 2002).

Carmel et al. (2001) proposed a pruning strategy where each indexed term is sent in turn as a query to their search system. Index information is discarded for those documents that contain the query term, but do not appear in the top ranked results in response to the query. This strategy is computationally expensive and time consuming. The soundness of this approach is unclear; the highly ranked pages for many queries are not highly ranked for any of the individual query terms, de Moura et al. (2005) have extended Carmel's method. The apply Carmel's method to extract the most important terms of each collection. Then they keep only those sentences that contain important terms, and delete the rest. For the same reason discussed previously, this approach also is not applicable in uncooperative DIR environments. Although this approach is more effective than Carmel's method in most cases, the loss in average precision compared to a full-text baseline is significant.

D'Souza, Thorn, and Zobel (2004) discuss surrogate methods for pruning, where only the most significant words are kept for each document. In this approach, the representation set is not the collection's vocabulary, but is instead a complete index for the surrogates. Such an approach requires a high level of cooperation between servers.

Craswell et al. (1999) use a pruning strategy to merge the results gathered from multiple search engines. In their work, they download the first four kilobytes of each returned document instead of extracting the whole document for result merging. They showed that in some cases, the loss in performance is negligible. They only evaluated their method for result merging.

In a comprehensive analysis of pruning in brokers, Lu and Callan (2002) divided pruning methods into various groups: *frequency-based* methods prune documents according to lexicon statistics; *location-based* methods exclude terms based on the position of their appearance in the documents; *single-occurrence* methods set a pruning threshold based on the number of unique terms in documents, and keep one instance of each term in the document; and *multiple-occurrence* methods allow for multiple occurrences of terms in pruned documents. Experiments evaluating the performance of nine methods demonstrated that four models can achieve similar optimal levels of performance, and do not have any significant advantage over each other.

Of these best methods, FIRSTM is the only one which does not rely on a broker's vocabulary statistics. For each document, this approach stores information about the first 1600 terms. The other methods measure the importance of terms based on frequency information. As discussed, these methods are unsuitable for DIR in many ways: the frequency of a term in the broker does not indicate its importance in the original collections; the cost of pruning and re-indexing might be high; and, adding a new collection makes the current pruned index unusable, since after a new collection is added to the system, the previous information is no longer valid. The FIRSTM approach prunes during parsing, which makes it more comparable with our approach. Therefore, we evaluate our approach by using FIRSTM as a baseline.

Lu and Callan tested their methods on 100 collections created from TREC disks 1, 2, and 3, and showed that their models can reduce storage costs by 54%–93%, with less than a 10% loss in precision. We test our systems on WT10 g and GOV1, which are larger and consist of unmanaged data crawled from the Web; these collections are described in more detail in Section 3.2.

Our proposed pruning method is applied during parsing, and is independent of index updates, as the addition of new collections to the system does not require the re-indexing of the original documents. Moreover, our pruning method does not reduce system performance and precision, while in all of the discussed previous work (Carmel et al., 2001; Craswell et al., 1999; de Moura et al., 2005; Lu & Callan, 2002), pruning results in a decrease in precision.

3.2. Using query logs for pruning

The main motivation for pruning is to omit unimportant terms from the index. That is, pruning methods are intended to exclude the terms that are less likely to appear in user queries (de Moura et al., 2005). Some methods prune terms that are rare in documents (Lu & Callan, 2002). However, the distribution of terms in user queries is not similar to that in typical web documents.

We propose using the history of previous user queries to achieve this directly. Our hypothesis is that pruning those terms that do not appear in a search engine query logs will be able to reduce index sizes while maintaining retrieval performance. We test our hypothesis with experiments on distributed environments and

central indexes for different types of queries. In a standard search environment, where completeness may be more important than improvements in efficiency, such pruning (or any pruning) is unappealing; but in a distributed environment, where index information is incomplete and is difficult to gather, such an approach has significant promise.

For our experiments, we used a list of the 315936 unique terms in the log of about one million queries submitted to the Excite search engine on 16 September 1997 (Spink et al., 2001). Larger query logs, or a combination of query logs from different search engines, might be useful for larger collections. Also, for highly topic-specific collections, topical query logs (Beitzel, Jensen, Chowdhury, Grossman, & Frieder, 2004) and query terms that have been classified into different categories (Jansen, Spink, & Pedersen, 2005) could provide additional benefits.

For experiments on uncooperative DIR environments and brokers, we used the testbed described in Section 2.3. The 100 largest servers were extracted from the TREC WT10 g collection, with each server being considered as a separate collection. Query-based sampling, as described in Section 2.1, was used to obtain representation sets for each collection by downloading 300 documents from each server in our testbed. We do not omit stopwords in any of our experiments. For each downloaded sample, we only retained information about those terms that were present in our query log, and eliminated the other terms from the broker. We used CORI (Callan et al., 1995) for collection selection and result merging; CORI has been used in many papers as a baseline (Craswell et al., 2000; Nottelmann & Fuhr, 2003; Powell & French, 2003; Si & Callan, 2003).

TREC topics 451–550 and their corresponding relevance judgements were used to evaluate the effectiveness of our pruned representation sets. We use only the ⟨title⟩ field of TREC topics as queries for the search system.

To test our pruning method on central indexes, we used the TREC WT10 g and GOV1 collections. The GOV1 collection (Craswell & Hawking, 2002) contains over a million documents crawled from the .gov domain. TREC topics 451–550 were used for our experiments on WT10 g, and topics 551–600 and NP01–NP150 were used with the GOV1 collection. All experiments with central indexes use the OkapiBM25 similarity measure (Robertson, Walker, Hancock-Beaulieu, Gull, & Lau, 1992).

In addition to these topic-finding search tasks, we evaluate our pruning approach on central indexes for named-page finding and topic distillation tasks. In *topic distillation*, the objective is to find relevant homepages related to a general topic (Craswell, Hawking, Wilkinson, & Wu, 2003). We use the TREC topic distillation topics 551–600, and corresponding relevance judgements, with the GOV1 collection. For *named-page finding* (also known as homepage finding) the aim is to find particular web pages of named individuals or organisations. To evaluate this type of search task, we used the TREC named-page finding queries NP01-NP150 (Craswell & Hawking, 2002).

3.3. Distributed retrieval results

The results of our experiments using different pruning methods for DIR systems are shown in Table 3. For each scheme, up to 1000 answers were returned per query. The cutoff (CO) values show the number of collections that are selected for each query. That is, for the first row, only the best collection is selected while for the

Table 3
Effectiveness of different pruning schemes, on a subset of WT10 g

CO	MAP			P@10			*R*-precision		
	ORIG	FIRSTM	PR	ORIG	FIRSTM	PR	ORIG	FIRSTM	PR
1	0.0178	0.0096	0.0178	0.0367	0.0316	0.0367	0.0217	0.0071	0.0217
10	0.0415	0.0399	0.0415	0.0620	0.0468	0.0630	0.0593	0.0400	0.0594
20	0.0355	0.0468	0.0504	0.0590	0.0494	0.0646	0.0485	0.0460	0.0644
30	0.0399	0.0462	0.0546	0.0603	0.0481	0.0658†	0.0500	0.0426	0.0659*
40	0.0489	0.0509	0.0628†	0.0641	0.0561	0.0671†	0.0521	0.0437	0.0611
50	0.0506	0.0516	0.0647†	0.0654	0.0565	0.0684†	0.0562	0.0439	0.0708*

Significance at the 0.1 and 0.05 levels of confidence are indicated with * and †, respectively. "CO" is the cutoff number of servers from which answers are fetched.

Table 4
Effectiveness of a complete index on WT10 g with TREC topics 451–550

MAP	P@5	P@10	*R*-precision
0.1338	0.1392	0.1139	0.1397

last row, the top 50 collections – half of the available collections - are selected. Our pruning method (PR), and original un-pruned index (ORIG), produce the same performance for cutoff values of 1 and 10 For larger cutoff values, the broker with pruned documents is able to select better collections, and outperforms the system that uses the original documents. The PR approach consistently outperforms the FIRSTM approach (described in Section 3.1), at all cutoff points. We tested our results using a paired *t*-test; significance at the 0.1 and 0.05 levels of confidence are indicated with * and †, respectively.

Overall, performance based on MAP is low because there are few relevant answers for the queries in the testbed. To measure system performance when all documents are available, we indexed all documents from all 100 servers using a central index. Results for all 100 collections, searched using a central index, are shown in Table 4.

3.4. Central index pruning results

The experiments reported in the previous section were undertaken in an uncooperative distributed environment. In this section, we evaluate our pruning method on central indexes, considering different search tasks. We evaluate the performance of five different pruning approaches.

Under the FIRSTM approach, only information about the first 1600 terms in each document is added to the index. For our PR scheme, we only store index information about terms that occur in our query log. We also test the effect of using fewer query log terms on performance. The PR(tf) approach only indexes terms that have occurred two or more times in the query log; all the other terms that occur in documents are pruned. This reduces the size of our query log from 315936 unique terms to 108425 terms.

We also investigate two hybrid models. The FIRSTM-PR scheme only retains information about terms that are present in the query log and also occur in the first 1600 words of each document. FIRSTM-PR(tf) restricts index information further, adding an additional constraint whereby terms need to occur two or more times in the query log to be indexed.

Table 5 shows the effectiveness of the various schemes for a topic-finding task on the WT10 g collection. The original, full-text index is labelled ORIG. The performance of the PR and PR(tf) schemes is similar to the original index. The *t*-test does not show any significant difference between the performance of the original index, PR, and PR(tf) at the 0.1 or 0.05 levels. The FIRSTM scheme harms performance compared to the full-text index. This reduction is significant at the 0.05 level for the MAP, P@5 and *R*-precision metrics. The hybrid models, FIRSTM-PR and FIRSTM-PR(tf), do not reduce precision below the FIRSTM scheme. However, the hybrid approaches are able to reduce index size further, as is discussed below.

The outcomes of using the five pruning approaches for topic distillation queries are shown in Table 6. All schemes are able to maintain retrieval performance compared to the full-text index ORIG. For P@5, the PR and

Table 5
Comparison of central index pruning schemes for topic-finding queries on WT10 g

	MAP	P@5	P@10	*R*-precision
ORIG	0.1730	0.3469	0.2929	0.2061
FIRSTM	0.1560†	0.3204†	0.2786	0.1921†
PR	0.1728	0.3490	0.2929	0.2057
PR(tf)	0.1708	0.3469	0.2918	0.2040
FIRSTM-PR	0.1574†	0.3265†	0.2796	0.1937†
FIRSTM-PR(tf)	0.1559†	0.3224†	0.2816	0.1922†

Significance at the 0.1 and 0.05 levels of confidence are indicated with * and †, respectively.

Table 6
Comparison of central index pruning schemes for topic distillation queries on GOV1

	MAP	P@5	P@10	*R*-precision
ORIG	0.1280	0.1673	0.1529	0.1470
FIRSTM	0.1280	0.1673	0.1529	0.1470
PR	0.1294	0.1918	0.1673	0.1540
PR(tf)	0.1294	0.1918	0.1673	0.1540
FIRSTM-PR	0.1253	0.1633	0.1612	0.1456
FIRSTM-PR(tf)	0.1256	0.1633	0.1633	0.1457

PR(tf) schemes are able to increase performance above the baseline. However, this increase is not statistically significant.

The results for the named-page search task are shown in Table 7. Many of these queries contain *out-of-vocabulary* terms, which are not present in the vocabulary of the index. Since our pruning strategy excludes any term which is not available in the query log, we were expecting to see a significant loss in performance. However, the *t*-test showed that none of the differences between the original index (ORIG) and our query log pruning scheme (PR) are statistically significant even at the 0.1 level. The results demonstrate that, on average, all systems are able to return the correct answer as the second document in the results list. Because the out-of-vocabulary problem is more prevalent for the query log schemes, the proportion of queries where no correct answer is found in the first 1000 items is slightly higher than for the other schemes.

As explained in Section 3.2, we use the Okapi BM25 similarity function (Robertson et al., 1992) for our central index effectiveness experiments. To check that the trends in our results are not an artifact of a particular similarity measure, we also investigated the effect of pruning using three different ranking functions on each collection: $tf \cdot idf$ (Salton & McGill, 1986); INQUERY (Callan, Croft, & Harding, 1992); and the KL-divergence language model (Lafferty & Zhai, 2001). The trends in results were similar for all ranking functions, and for brevity are not included here.

One of the key aims of index pruning is to reduce the amount of data that needs to be maintained, either in a central index, or in the document representation sets used by a broker in distributed IR. The effect of the various pruning schemes on the size of indexes is shown in Table 8. For the WT10 g collection, the FIRSTM, PR, and PR(tf) schemes all reduce the size of index by about 30%. The hybrid models, FIRSTM-PR and FIRSTM-PR(tf), are able to reduce index size by an additional 15%. As demonstrated by our effectiveness results,

Table 7
Comparison of central index pruning schemes for named-page finding queries on GOV1

	MRR	% Top 1	% Top 10	% Fail
ORIG	0.5099	0.3647	0.7466	0.0353
FIRSTM	0.5099	0.3647	0.7466	0.0353
PR	0.4818	0.3294	0.7400	0.0588
PR(tf)	0.4619	0.3117	0.7066	0.0647
FIRSTM-PR	0.4725	0.3294	0.7066	0.0411
FIRSTM-PR(tf)	0.4619	0.3117	0.6800	0.0470

Table 8
The effect of pruning on index size, measured in gigabytes

	WT10 g		GOV1	
ORIG	2.58	(100%)	3.15	(100%)
FIRSTM	1.77	(68.6%)	1.80	(57.1%)
PR	1.88	(72.8%)	2.27	(72.0%)
PR(tf)	1.81	(70.1%)	2.18	(69.2%)
FIRSTM-PR	1.42	(55.0%)	1.36	(43.1%)
FIRSTM-PR(tf)	1.38	(53.4%)	1.32	(41.9%)

Table 9
The effect of pruning on broker size, measured in megabytes

ORIG	160	(100%)
FIRSTM	58	(36.2%)
PR	125	(78.1%)
PR(tf)	115	(71.8%)
FIRSTM-PR	48	(30.0%)
FIRSTM-PR(tf)	46	(28.7%)

these schemes are differentiated by search performance, FIRSTM reduces topic finding performance, while PR and PR(tf) do not alter it from the full-text baseline. Moreover, while the FIRSTM and hybrid models result in similar levels of precision, the hybrid models reduce the index size by an additional 15% over FIRSTM. Therefore, if search performance is to be optimised, the PR scheme is to be preferred. If efficiency is of greater concern, one of the hybrid models should be chosen.

There is more variation among the efficiency gains for the various schemes when they are applied to the GOV1 collection. Here, the PR and PR(tf) schemes show the least reduction in index size, reducing the space required by around 30%. The FIRSTM approach is able to give a reduction of 43%, while the hybrid models reduce the index size by about 57%. In terms of effectiveness, the schemes showed similar levels of accuracy for topic distillation and named-page finding tasks on the GOV1 collection. For these search tasks we therefore recommend the use of the hybrid schemes, which minimise index size while retaining performance effectiveness.

Efficiency results for our distributed testbed are shown in Table 9. Here, FIRSTM achieves a reduction index size of 64%, which is substantially greater than the 22–28% reduction given by the PR and PR(tf) schemes. However, there is again a tradeoff with effectiveness, since the FIRSTM approach often leads to decreases compared to the PR scheme. As in the central index case, the hybrid models are able to reduce index size slightly further compared to FIRSTM.

4. Conclusion

We have proposed two new applications of query logs for the improvement of distributed information retrieval systems. The first is a novel sampling approach for distributed collections in an uncooperative environment. Our approach focused the sampling process by using terms that are available in search engine query logs. Our experiments demonstrated that the method is no more costly than previous approaches to query-based sampling, but produces samples that allow retrieval to be significantly more effective.

The second application we propose is that query logs can be used to focus index pruning strategies towards terms that are important to users. Our pruning strategy is able to maintain system effectiveness compared to a full-text index, while being able to reduce index size by 22–28%. We evaluated our approach for various web search tasks, including topic distillation and named-page finding. Although many of these topics contain out-of-vocabulary terms, the pruned indexes retrieve relevant answers as effectively as the original index. We compared our strategies with the FIRSTM pruning approach, which can give additional savings in index size but often results in decreased search effectiveness. We also proposed two hybrid models that combine features of the FIRSTM and query-log approaches, and showed that such models can further reduce index size, while not reducing effectiveness below the FIRSTM-only scheme. For standard retrieval applications, such pruning may not be desirable, but in environments such as DIR where pruning may be essential such an approach is an effective choice.

While our results demonstrate that using query logs can be an effective mechanism to guide both collection sampling and index pruning, it is well-known that query topics shift over time. As future work, we plan to simulate and evaluate the effects of these changing patterns on the effectiveness of our schemes. We also plan to investigate the effectiveness of our approaches on larger collections, such as the TREC GOV2 collection. Finally, we plan to investigate the effects of combining query logs from different search engines, and whether this can further enhance the robustness of our techniques. However, our results already clearly demonstrate that query logs provide a robust basis for DIR in uncooperative environments, with significant improvements in effectiveness compared to previous methods.

References

Bailey, P., Craswell, N., & Hawking, D. (2003). Engineering a multi-purpose test collection for Web retrieval experiments. *Information Processing and Management, 39*(6), 853–871.

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D., & Frieder, O. (2004). Hourly analysis of a very large topically categorized Web query log. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 321–328). Sheffield, United Kingdom: ACM Press.

Billerbeck, B., Scholer, F., Williams, H. E., & Zobel, J. (2003). Query expansion using associated queries. In O. Frieder, J. Hammer, S. Quershi, L. Seligman (Eds.), *Proceedings of the CIKM international conference on information and knowledge management* (pp. 2–9). New Orleans.

Callan, J., & Connell, M. (2001). Query-based sampling of text databases. *ACM Transactions on Information Systems, 19*(2), 97–130.

Callan, J., Connell, M., & Du, A. (1999). Automatic discovery of language models for text databases. In *Proceedings of the 1999 ACM SIGMOD international conference on management of data* (pp. 479–490). Philadelphia, Pennsylvania: ACM Press.

Callan, J., Croft, W. B., & Harding, S. M. (1992). The INQUERY retrieval system. In *Proceedings of third international conference on database and expert systems applications* (pp. 78–83). Valencia, Spain.

Callan, J., Lu, Z., & Croft, W. B. (1995). Searching distributed collections with inference networks. In *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 21–28). Seattle, Washington: ACM Press.

Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y. S., & Soffer, A. (2001). Static index pruning for information retrieval systems. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 43–50). New Orleans, Louisiana: ACM Press.

Craswell, N., & Hawking, D. (2002). Overview of the TREC-2002 Web Track. In *Proceedings of TREC-2002*. Gaithersburg, Maryland.

Craswell, N., Bailey, P., & Hawking, D. (2000). Server selection on the World Wide Web. In *Proceedings of the fifth ACM Conference on digital libraries* (pp. 37–46). San Antonio, Texas: ACM Press.

Craswell, N., Hawking, D., Wilkinson, R., & Wu, M. (2003). Overview of the TREC-2003 Web Track. In *Proceedings of TREC-2003*. Gaithersburg, Maryland.

Craswell, N., Hawking, D., & Thistlewaite, P. (1999). Merging results from isolated search engines. In *Proceedings of the 10th Australasian database conference* (pp. 189–200). Auckland, NZ: Springer-Verlag.

Cui, H., Wen, J., Nie, J., & Ma, W. (2002). Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web* (pp. 325–332). Honolulu, Hawaii: ACM Press.

de Moura, E., dos Santos, C. F., Fernandes, D. R., Silva, A. S., Calado, P., & Nascimento, M. A. (2005). Improving Web search efficiency via a locality based static pruning method. In *Proceedings of the 14th international conference on World Wide Web* (pp. 235–244). Chiba, Japan: ACM Press.

D'Souza, D., Thorn, J., & Zobel, J. (2004). Collection selection for managed distributed document databases. *Information Processing and Management, 40*(3), 527–546.

Fagni, T., Perego, R., Silvestri, F., & Orlando, S. (2006). Boosting the performance of Web search engines: Caching and prefetching query results by exploiting historical usage data. *ACM Transactions on Information Systems, 24*(1), 51–78.

Fuhr, N. (1999). A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems, 17*(3), 229–249.

Gravano, L., Chang, C. K., Garcia-Molina, H., & Paepcke, A. (1997). STARTS: Stanford proposal for Internet meta-searching. In *Proceedings of the 1997 ACM SIGMOD international conference on management of data* (pp. 207–218). Tucson, Arizona: ACM Press.

Gravano, L., Garcia-Molina, H., & Tomasic, A. (1999). GlOSS: Text-source discovery over the Internet. *ACM Transactions on Database Systems, 24*(2), 229–264.

Gravano, L., Ipeirotis, P. G., & Sahami, M. (2003). Qprober: A system for automatic classification of Hidden-Web databases. *ACM Transactions on Information Systems, 21*(1), 1–41.

Hoi, C., & Lyu, M. R. (2004). A novel log-based relevance feedback technique in content-based image retrieval. In *Proceedings of the 12th annual ACM international conference on multimedia* (pp. 24–31). New York, NY: ACM Press.

Ipeirotis, P. G., & Gravano, L. (2002). Distributed search over the Hidden Web: Hierarchical database sampling and selection. In *Proceedings of 28th international conference on very large data bases* (pp. 394–405). Hong Kong, China.

Jansen, B. J., & Spink, A. (2005). An analysis of Web searching by European AlltheWeb.com users. *Information Processing and Management, 41*(2), 361–381.

Jansen, B. J., Spink, A., & Pedersen, J. (2005). A temporal comparison of AltaVista Web searching: Research articles. *Journal of the American Society for Information Science and Technology, 56*(6), 559–570.

Jones, K. S., & Rijsbergen, C. V. (1976). Progress in documentation. *Journal of Documentation, 32*(1), 59–75.

Lafferty, J., & Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 111–119). New Orleans, Louisiana: ACM Press.

Lu, J., & Callan, J. (2002). Pruning long documents for distributed information retrieval. In *Proceedings of the eleventh international conference on information and knowledge management* (pp. 332–339). McLean, Virginia: ACM Press.

Nottelmann, H., & Fuhr, N. (2003). Evaluating different methods of estimating retrieval quality for resource selection. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 290–297). Toronto, Canada: ACM Press.

Persin, M., Zobel, J., & Sacks-Davis, R. (1996). Filtered document retrieval with frequency-sorted indexes. *Journal of the American Society of Information Science, 47*(10), 749–764.

Powell, A. L., & French, J. (2003). Comparing the performance of collection selection algorithms. *ACM Transactions on Information Systems, 21*(4), 412–456.

Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gull, A., & Lau, M. (1992). Okapi at TREC. In *Proceedings of TREC-1992* (pp. 21–30). Gaithersburg, Maryland.

Salton, G., & McGill, M. (1986). *Introduction to modern information retrieval*. New York, NY, USA: McGraw-Hill, Inc.

Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: Effort, sensitivity, and reliability. In *Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 162–169). Salvador, Brazil: ACM Press.

Shokouhi, M., Scholer, F., & Zobel, J. (2006). Sample sizes for query probing in uncooperative distributed information retrieval. In X. Zhou, J. Li, H. T. Shen, M. Kitsuregawa, & Y. Zhang (Eds.). *APWeb. Lecture Notes in Computer Science* (vol. 3841, pp. 63–75). Springer.

Si, L., & Callan, J. (2003). Relevant document distribution estimation method for resource selection. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 298–305). Toronto, Canada: ACM Press.

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

Spink, A., Wolfram, D., Jansen, B., & Saracevic, T. (2001). Searching the Web: The public and their queries. *Journal of the American Society for Information Science and Technology, 53*(2), 226–234.

Voorhees, E. M., & Harman, D. (2000). Overview of the sixth Text REtrieval Conference (TREC-6). *Information Processing and Management, 36*(1), 3–35.

Wen, J., Lao, N., & Ma, W. (2004). Probabilistic model for contextual retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 57–63). Sheffield, United Kingdom: ACM Press.

Xu, J., & Callan, J. (1998). Effective retrieval with distributed collections. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 112–120). Melbourne, Australia: ACM Press.

Yuwono, B., & Lee, D. L. (1997). Server ranking for distributed text retrieval systems on the Internet. In *Proceedings of the fifth international conference on database systems for advanced applications* (pp. 41–50). Melbourne, Australia: World Scientific Press.