Updating Collection Representations For Federated Search

Milad Shokouhi
School of Computer Science and Information Technology
RMIT University
Melbourne, Australia
milad@cs.rmit.edu.au

Mark Baillie
Department of Computer and Information Sciences
University of Strathclyde
United Kingdom G1 1XH
mb@cis.strath.ac.uk

Leif Azzopardi
Department of Computing Science
University of Glasgow
United Kingdom G12 8QQ
leif@dcs.gla.ac.uk

ABSTRACT

To facilitate the search for relevant information across a set of online distributed collections, a federated information retrieval system typically represents each collection, centrally, by a set of vocabularies or sampled documents. Accurate retrieval is therefore related to how precise each representation reflects the underlying content stored in that collection. As collections evolve over time, collection representations should also be updated to reflect any change, however, a current solution has not yet been proposed. In this study we examine both the implications of out-of-date representation sets on retrieval accuracy, as well as proposing three different policies for managing necessary updates. Each policy is evaluated on a testbed of forty-four dynamic collections over an eight-week period. Our findings show that out-of-date representations significantly degrade performance over time, however, adopting a suitable update policy can minimise this problem.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Distributed Systems; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*hidden web*

General Terms

Design, Experimentation

Keywords

federated search, distributed information retrieval, collection selection

1. INTRODUCTION

There has been a recent trend towards the investigation of alternative solutions for accessing online content that cannot be readily accessed through standard means, such as content crawling or harvesting, often referred to as the *hidden-web* [Price and Sherman, 2001]. Federated search systems [Avrahami et al., 2006], also known as distributed information retrieval [Callan, 2000] or selective meta-searchers [Craswell et al., 2004], have recently been proposed as a solution to accessing and searching content found in the hidden-web. An open problem that federated search systems face is how to represent these large document repositories and databases (i.e. *collections*) both accurately and efficiently to ensure effective retrieval performance; because a collection representation that is not reflective of the underlying content will have a negative impact on the accuracy of a search.

In this paper we focus on the problem of maintaining representation sets for dynamically changing, uncooperative, distributed collections. Previous research into this problem has largely worked under the implicit assumption that these collections are static. However, over time the content may have been updated and modified considerably, new content may have been added to the collection or old content deleted. Without an effective updating strategy, collection representations become out-of-date, which can impact on retrieval accuracy – as shown in Section 3. The aim of this study is to examine the problem of managing dynamic collections for federated search, and in particular how to effectively update collection representation sets. Three updating policies are investigated; (i) updating all available representation sets equally with the same rate, (ii) attempting to identify the *popular* collections, and updating collections according to their *popularity*, and (iii) by updating the representation sets according to the estimated collection sizes i.e. the representation of larger collections are updated more frequently.

We first describe an existing problem by showing the impact of using out-of-date representation sets on the final search precision. Then, we introduce the three updating methods and show that they can significantly improve the search effectiveness compared to a baseline scenario (no updating strategy).

2. BACKGROUND

The aim of a federated information retrieval (FIR) system is to provide a search service over non-crawlable collections through the means of a centralised (search) broker [Callan, 2000]. To achieve this objective, a FIR system has to address a number of issues such as (i) the acquisition and maintenance of collection representation sets, (ii) collection selection, the problem of identifying and searching only the subset of collections that contain relevant documents with

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

respect to the user query, and (iii) results merging, the process of gathering, merging and then producing an accurate ranked list of results to present to the user.

The broker manages the retrieval process and co-ordinates the phases within the FIR system. To facilitate collection selection and results merging, the broker stores locally a representation set of documents and vocabularies from each collection. Representation sets can be generated in a variety of forms depending on a number of influencing factors such as the retrieval model used for collection selection, and the level of cooperation between search service and information provider. Currently adopted representations include a term vector of counts (i.e. word histogram) or probabilities (i.e. a language model) [Callan and Connell, 2001], a sample of indexed documents from each collection [Si and Callan, 2004], a hierarchical topical summary [Gravano et al., 2003], or the full collection index [Callan, 2000]. For cooperative collections, content statistics can be accessed through an agreed protocol such as STARTS [Gravano et al., 1997] or SDLIP [Paepcke et al., 2000]. However, when cooperation cannot be guaranteed – that is, collections are uncooperative – other means are required such as *query-based sampling* [Callan and Connell, 2001] or *focused probing* [Gravano et al., 2003].

During query-based sampling, an estimated representation is generated by submitting random queries to the collection, incrementally adding the newly retrieved documents to the estimated representation set [Callan and Connell, 2001]. Queries are randomly chosen from the retrieved content or an external vocabulary, such as query logs [Craswell et al., 2000; Shokouhi et al., 2007], to ensure that an unbiased sample estimate has been obtained. In contrast, focused-probing submits topically related terms to the collection in order to obtain *biased* representation sets that are topically focused [Gravano et al., 2003]. The result is a biased representation set based on the topic along with the categorisation of the database. Sampling is terminated for both approaches when a sufficiently good representation of the underlying collection has been acquired which facilitates accurate retrieval [Avrahami et al., 2006; Baillie et al., 2006; Shokouhi et al., 2007].

There have been a number of proposed approaches to collection selection that can be grouped into two main categories. The first family of techniques are analogous to document retrieval. That is, an index is formed from the union of all representation sets for each collection, where the representation set is treated as a bag of words. Given a user query, the representation sets are then ranked in order of relevance, with a subset of the top ranked collections then searched. Techniques differ by how the representation sets are ranked, for example, using Bayesian inference network (CORI) [Callan et al., 1995], vector space model (GLOSS) [Gravano et al., 1999], or language models [Xu and Croft, 1999; Si et al., 2002]. However, the decision to remove documents boundaries within the representation set is thought to impact on collection selection performance [Xu and Callan, 1998], with a number of recent empirical studies supporting this claim [Si and Callan, 2003a; Hawking and Thomas, 2005]. As a consequence, a new group of techniques have been proposed that retain the document boundaries within the representation sets, such as ReDDE [Si and Callan, 2003a], CRCS [Shokouhi, 2007], UUM [Si and Callan, 2004], HARP and AWSUM [Hawking and Thomas, 2005]. A common theme shared across these techniques is that the representation sets are combined to form a sampled centralised index. Thus, given a user query, documents in the sampled index are ranked. This document ranking is then used to predict which collections have the largest distribution of relevant resources that is then utilised as a decision process for selecting the subset of collections to search. We use CRCS for our experiments in this paper, as it has been suggested to be more robust than current methods in the absence of training queries [Shokouhi, 2007].

Finally, the returned documents from each collection are merged and then ranked. The widely adopted techniques for results merging are CORI merge [Callan et al., 1995] and SSL [Si and Callan, 2003b]. Both approaches normalise the relevance scores from the retrieved documents to enable merging and ranking, although SSL has been found to be more robust and effective [Si and Callan, 2003b]. The CORI merge algorithm normalises the document scores across each collection and combines them with the collection ranking scores using a linear combination to provide the final merged document ranking. Alternatively, SSL uses linear regression to re-estimate the relevance of each document based on the document and collection scores. Both approaches utilise statistics obtained from the collection representation sets to estimate parameters required for merging.

The entire FIR process, defined by the combination of these approaches, has largely been tested under the implicit premise that collections are static. As the generation of accurate representation sets is important during each phase of FIR [Avrahami et al., 2006; Baillie et al., 2006; Shokouhi et al., 2007], we further hypothesise that maintaining these representations is also imperative; out-of-date representations will invariably have a negative impact on collection selection accuracy, parameter estimation during results merging, and ultimately retrieval performance.

3. DYNAMIC CONTENT AND UPDATES

Dynamic collections can change in a variety of ways. Documents may be added or deleted from the collection, as well as the updating of content within existing documents. In the context of centralised IR, such as web search engines, the *freshness* of data is important as the search engine usually reflects the user's perception of the web [Craswell et al., 2004]. If content is out-of-date or missing, this can have a negative impact on how the user perceives the search engine. In order to ensure fresh data and reduce inconsistencies between index and documents, indexes are constantly updated using predefined revisit policies [Cho and Garcia-Molina, 2003]. For example, a uniform policy would involve the crawler updating documents from all sites independent of that rate of change at each site. A non-uniform policy, in comparison, would involve the crawler revisiting some collections more often than others based on predefined criteria, such as how frequent a collection updates. Cho and Garcia-Molina [2003] discovered that the choice of criteria can have an impact on both system resources and index quality, where adopting a non-uniform policy based on the frequency of change is not necessarily an optimal solution in comparison to a uniform policy, especially if the changes within a collection are too frequent.

In comparison, the challenges of dynamic collections differ in the context of FIR. An assumed advantage of FIR over centralised IR is that the problem of fresh data is minimised.

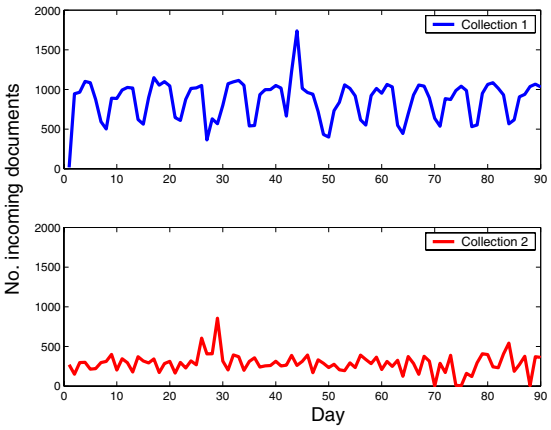

Figure 1: The frequency of new documents per day for two news collections.

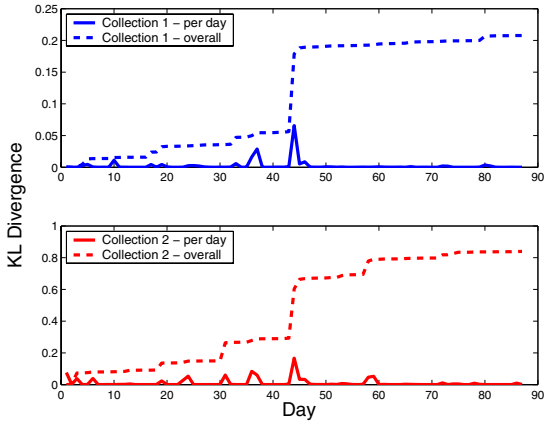

Figure 2: The change in representation set vocabulary per day for two news collections.

This is based on the assumption that if the local collections update the corresponding search index on the arrival of new documents, then every time a FIR system correctly routes a query to the subset of relevant collections, the latest available content will be searched. However, given this advantage, collection representation sets are still required to be up-to-date in order to ensure accurate collection selection. Ipeirotis et al. [2005] illustrated how the vocabulary contained in a representation set of a dynamic collection deteriorates over time when it is not maintained through periodic updates. We show for the first time that retrieval accuracy will also deteriorate if the content within a collection has changed substantially from what is reflected in the representation set.

3.1 Changes in dynamic collections

To illustrate the changes in document size and vocabulary in dynamic collections we consider collections from two news agencies. Collection 1 contains up-to-date news reports concerning global current affairs, while the smaller Collection 2 focused on European sports and entertainment news. Content held within both collections was constantly downloaded over a 90 day period from April to June 2006.

The plots in Fig. 1 present the number of new incoming documents that are added to each collection daily. Collection 1 has a similar weekly trend of new documents that peak during the midweek and tail off towards the weekend. In comparison, the number of updates in Collection 2 does not follow such a consistent trend. Collection 2 receives less documents per day than Collection 1, with less fluctuations across the week. During major news events, the increase in average documents increased substantially from the norm for both collections, such as on day 44 for Collection 1 and days 27 and 29 for Collection 2.

In Fig. 2, the plots illustrate that the arrival of new documents effects the representation set vocabulary for each collection. This is shown by the Kullback-Leibler (KL) divergence between the term probability distributions (i.e language models) of the representation set of the previous and current day (solid line) [Kullback, 1959]. For each day, we used the full information of the collection to form the representation set. We also show the KL divergence between the original representation set on day 1 against the subsequent representation sets of the following days (dashed line). The KL divergence represents how far the vocabularies of each representation set differs, where a distance of zero indicates no change.

For Collection 1, in general, the daily arrival of new documents has minimal effect on the representation set. However, during periods of breaking stories, there is a sharp change in the representations sets from the previous to the current day. These bursts of activity change the vocabulary both with new terms added to the vocabulary, and also the collection frequency of certain terms changing, which in turns results in a swap in term ranking. These subtle changes over time have a sustained impact on the representation set, with the difference between the representation set on day 1 and the following days becoming gradually further apart (the dashed line). In comparison, the representation sets of Collection 2 are affected more by the arrival of new documents. This could be a reflection of the both the size and content stored in Collection 2, where insertions result in larger changes in the collection vocabulary. These changes are noticeable both on a daily basis, and gradually over time. The overall result is a substantial difference between the vocabulary of the original representation set and the proceeding days. This is consistent with the experiments reported by Ipeirotis et al. [2005] on real web collections.

These changes in vocabulary of the representation sets can be thought of as a data stream [Kleinberg, 2006], where over a period of time some terms rise and fall in usage, and in particular there are "bursts" of activity where a term is commonly used, coinciding with breaking news stories. In other words, some terms and phrases are topical for short periods of time which is reflected in the changes in KL between representation sets.

3.2 Impact on retrieval accuracy

We have shown that the difference in vocabulary of representation sets of dynamic collections gradually deteriorates (i.e. does not reflect the underlying collection content) over time if the representation sets are not constantly updated. But what impact does this have on FIR performance?

Evaluation testbed. The standard testbeds for FIR are based on static TREC collections that do not specifically facilitate the evaluation of dynamic content [Callan, 2000].

Table 1: The crawling statistics for the pages downloaded from abc.com.au over eight weeks in 2004.

Crawl	Documents	Size(Mb)	Date
1	485 190	6 522	27 May
2	465 684	6 440	04 June
3	475 451	6 481	11 June
4	563 670	6 799	18 June
5	611 819	7 751	27 June
6	614 393	7 833	05 July
7	546 630	6 850	22 July
8	540 530	6 871	30 July

Table 2: The document statistics for the largest 44 collections downloaded from abc.com.au.

Crawl	Min	Average	Max
1	4	9 138	63 033
2	4	8 719	59 223
3	4	8 919	63 033
4	23	10 613	63 150
5	1 421	11 250	63 433
6	1 421	11 257	63 433
7	23	10 227	63 135
8	23	10 088	63 135

While these testbeds could be modified to consider time by assuming the documents are streamed by date, existing documents are never deleted nor are they updated. Therefore, to evaluate the impact of dynamic content we constructed a testbed derived from the Australian Broadcasting Corporation's (ABC) domain. The testbed is comprised of documents downloaded from the abc.net.au domain between May–July 2004. This also includes pages such as those starting with shop.abc.net.au. Documents are downloaded in eight separate crawl sets as shown in Table 1. We used a log of queries submitted to the abc.net.au domain during the period between 24 May–1 August 2004 for our evaluations. In total, there are 814 257 queries.

The crawled domains were broken into the sub domains according to their first directory level. Pages in each sub domain form a single collection. For our experiments, we only consider those sub domains that their total number of unique documents downloaded over the eight crawls were more than one thousand. This threshold left us with 44 collections of varying sizes for each crawl. Table 2 presents some information about the size of these collections for each crawl. The advantage of using this testbed is that between each crawl, content from some collections will have been updated, as well as new content inserted and old content deleted. For example, between crawls 2–6 the number of new documents added to the domain is more than the documents removed, with a net increase, while at crawl 7 and 8 the net effect is a decrease in the number of documents.

Oracle standard comparison. The effectiveness of FIR systems can be compared with that of a centralised system that has indexed the most recent version of documents. Since we only have the query logs that users issued during the period of the crawl, and not the corresponding relevance judgements or click through data, we adopted a different approach to evaluation. In a centralised IR model, all information about all the documents is known to the retrieval engine and so an *optimal* ranking is produced. In a federated model, the information that each retrieval engine has is limited. So the goal of our evaluation is to determine how well the techniques used in the federated model compared to the centralised approach [Craswell et al., 2000; Xu and Callan, 1998; Xu and Croft, 1999], which we shall refer to as the *oracle standard*. In other words, the ranking provided by the centralised model for the test query set are used as pseudo-relevance judgements in order to evaluate the different FIR techniques. For the oracle standard baseline we used OKAPI BM25 [Robertson et al., 1992] to rank all documents in the testbed based on a central index for each crawl. The top 100 ranked documents for each query were assumed to be relevant.

Experimental setup. Our aim was to investigate whether updating of content affected the performance of the FIR system i.e. is the system robust to such change.

We randomly selected 200 training queries from those submitted to the abc.com.au website on 24 May 2004. The selected queries are submitted three days earlier than our first crawl and do not overlap with the queries used in our testing experiments.

We used query-based sampling to gather the collection representation sets [Callan and Connell, 2001]. The *probe* queries were randomly selected from the content of sampled documents from collections. The sampling process for a collection was terminated after gathering 100 documents or sending 1 000 probe queries, whichever comes first. For each week, we used CRCS [Shokouhi, 2007] for collection selection. Avrahami et al. [2006] suggested that selecting 3–5 collections is usually sufficient for producing effective merged results. Therefore, we designed our experiments with two cutoff (CO) values ($CO \in \{3, 5\}$). We then applied SSL for ranking the returned answers from the selected collections and merging the results [Si and Callan, 2003b][1].

We repeated the experiment for the crawl sets 2–8 using the collection samples generated in the first set. That is, collections are selected according to their old representation sets gathered in the first week. The psuedo-relevance judgements for the oracle standard baseline were updated to reflect content changes after each crawl. After each crawl, we measured the P@5 and P@10 for both old and fresh representation sets using the pseudo-relevance judgements. Significance of differences between both old and fresh representations was tested using the paired T-test.

Results. Figure 3 compares the precision values obtained by running the queries on up-to-date (fresh) and out-of-date (old) representation sets. P@5 and P@10 show the precision values for the top 5 and top 10 documents in the merge list across the 200 test topics. The dashed lines in these figures represent the precision values obtained by running the queries on the fresh samples. That is, samples are downloaded from the documents in the same crawl. However,

[1]Similar trends were found when using other collection selection methods such as CORI [Callan et al., 1995]. For brevity, we do not report the results here. In this experiment we were interested in analysing what the impact out-of-date representation sets had on retrieval performance, therefore, our conclusions were not dependent on the choice of collection selection or merging algorithms.

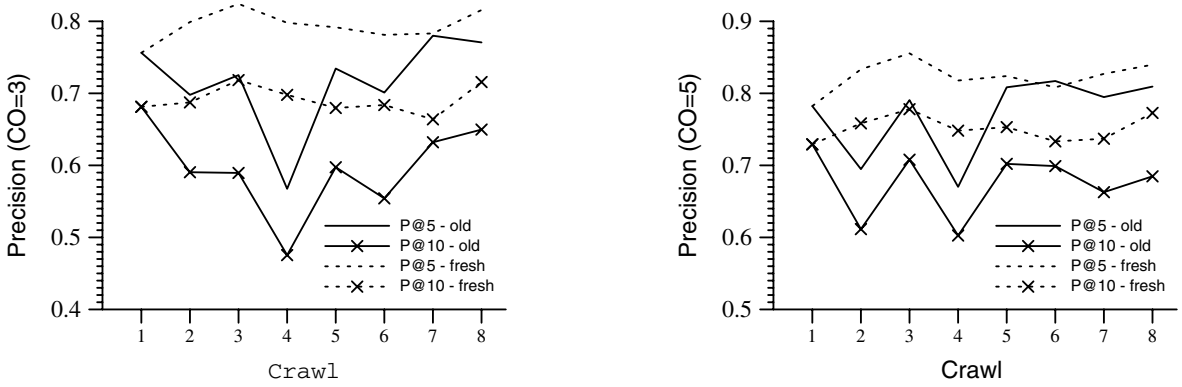

Figure 3: The impact of using out-of-date collection representation sets, on the final search effectiveness for 200 training queries. The CO values show the number of collections that are selected per query. Fresh and old respectively represent the results obtained by using the up-to-date and out-of-date representation sets.

for the curves specified with solid lines (old), collections are selected using the samples provided in the first crawl.

For the fresh representation sets, the precision values remain relatively constant, and report higher precision values on average than when using the old representation sets for both cutoff thresholds. In comparison, when using out-of-date representation sets, the precision values were inconsistent. For crawls 2–5, using fresh representations significantly improved over the old representation sets in terms of P@5 and P@10 across both cutoff values ($p < 0.01$). However, the out-of-date representation sets generated from collections during the first week (27 May) become *representative* for the fifth crawl (27 June) and the last crawl set (30 July). In other words, after a drop in precision relative to the fresh representation sets, there is a gradual increase in performance again. This improvement may be a result of a number of factors such as the monthly-update of some documents, "bursts" in popularity of certain vocabulary, or possibly changes in user query habits. For example, queries can be topical or even date dependent following trends related to events i.e. Breaking news events, the advent of SIGIR deadlines, Olympic games, etc. This relates back to the previous discussion that a representation set for a collection should reflect these trends where a term or phrase is commonly used. However, these bursts may only be temporary, hence the improvement in precision of the out-of-date representations.

The gaps between the effectiveness of old and fresh representation sets are usually significant for the late crawls. For example at crawl 6, the precision values obtained by using the fresh representation sets are significantly higher than the old samples for cutoff=3 ($p < 0.01$), and the P@10 values for fresh samples at crawl 8 are significantly better than the old ones ($p < 0.05$). Consequently, the consistency of performance is affected if the representation sets do not reflect such trends.

Summary. We showed the negative impact that out-of-date representation sets can have on FIR performance. It is therefore important to maintain and update representation sets periodically. However, the maintenance of an information resource presents many research challenges, particularly in uncooperative environments. In the following sections, we introduce and then evaluate three updating policies.

4. UPDATING METHODS

We now describe three updating methods for dynamic distributed collections. The first approach is a uniform policy that updates all available representation sets equally with the same rate. The remaining two policies are non-uniform, where some collections are updated more frequently than others based on a predefined criterion. One approach attempts to identify the *popular* collections, and updates collections according to their *popularity*. The other method simply updates the representation sets according to the estimated collection sizes. That is, the representation of larger collections are updated more frequently.

Notations. Describing each update policy requires some notation and definitions. For a given collection C, we assume that there is a corresponding representation set θ_C. This representation set is constructed from the set of documents previously sampled from the collection, $D_C = \{d_1, \ldots, d_n\}$. This defines the representation set at time 0, i.e. θ_C^0 using the corresponding document set, D_C^0. Thus, the representation set at any time t is defined by θ_C^t which uses the document set D_C^t. When no updating is applied, the representation set does not change over time and $\theta_C^t = \theta_C^0$. This is the standard assumption which we shall employ as a naive baseline in our experiments.

At each time step, it is necessary to decide whether or not the collection representations should be updated, and if so, how many documents should be sampled from each collection. Here, we focus on the latter concern and fix the time of updating. One of the considerations that must be made in the process of updating is to balance the efficiency with effectiveness. That is, it is desirable to restrict the number of documents that are sampled from each collection, in order to minimise costs. However, an appropriate number of documents need to be sampled from each collection to ensure that effectiveness is maintained. For these updating methods, we assume that a total fixed number of documents will be sampled at each time step from the collections, and that a proportion of these will be sampled from a particular collection. This holds the costs fixed and is defined by setting n equal to the total number of documents that we can afford to sample from the collections. The number of documents sampled at each time step for a particular collection is defined by n_C which is a proportion on n, defined by the

updating method. The document set at time t consists of the document set at time $t-1$ plus the new n_C sampled documents from collection C. This is then used to form the representation set θ_C^t at time t as below:

$$D_C^t = D_C^{t-1} + \{d_1^t, \ldots, d_{n_C}^t\} \quad (1)$$

If a new sampled document already exists in a collection representation set, it is replaced by the recent version.

Constant updates (CU). A simple way of updating collection representation sets is to distribute the number of documents we can afford to sample evenly between each collection i.e.

$$n_C = \frac{n}{N} \quad (2)$$

where the number of available collections is represented with N. If there are 44 collections and $n = 4\,400$, all representation sets are expanded by 100 documents at each time step. However, this is a naive approach as it assumes that all contents are both of uniform size and uniformly changing.

Updating according to query-logs (QL). Instead of evenly updating the collection, we posit that the collections that are more popular should be updated more than those that are not. By popular, we mean, those collections which are returned more often in response to the past queries issued by the collection selection algorithm (i.e. the highly ranked collections). That is because they are more likely to contain documents that satisfy the users' information needs. Hence, we use query logs to determine the popularity of collections and the number of documents to be sampled for a collection will be proportional to its popularity. So, at a given time t and for a collection C, the number of new n_C documents that are added to the document set D_C^{t-1} is:

$$n_C = \frac{n \times \sum_{q=1}^{Q} \rho_t(q, C)}{\sum_{i=1}^{N} \sum_{q}^{Q} \rho_t(q, C_i)} \quad (3)$$

where, Q is the total number of queries that are used for measuring the popularity of collections and $\rho_t(C, q)$ represents the rank of collection C for the training query q calculated by a collection selection method.

Updating according to collection sizes (SS). As opposed to using popularity, which may tend to a local optima because collections ranked high initially will tend to be favoured, we consider updating proportional the collection size. Here, we assume that larger collections will require more updating because they will potentially contain more relevant content and change more often. This can be formalised as:

$$n_C = n \times \frac{S_t(C)}{\sum_{i=1}^{N} S_t(C_i)} \quad (4)$$

where, $S_t(C)$ is the estimated size of collection C at time t. In our experiments, we use the start-of-the-art capture-history [Shokouhi et al., 2006] method to estimate the size of uncooperative collections.

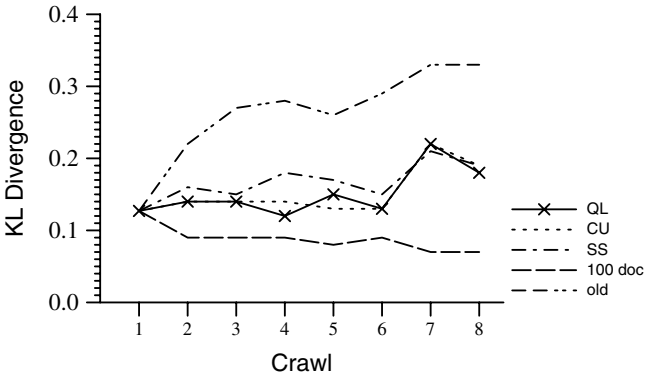

Figure 4: The KL divergence values produced by comparing the language models of representation sets with that of original collections. Numbers are median values over the 44 collections in each crawl.

5. EXPERIMENTAL RESULTS

We used 448 testing queries randomly selected from the ABC query log (see Section 3.2) to evaluate the updating methods. We stored the documents downloaded in each crawl separately in an oracle index. Pseudo-relevance judgements for each crawl are provided by running the testing queries on its oracle index. In the first crawl set, we used query-based sampling to generate the collection representation sets. We terminated sampling after downloading 100 documents or sending 1 000 probe queries (whichever comes first). The QL updating method requires some data from the previous crawls to calculate the number of documents that should be added to each representation set. Therefore, we start comparing the methods from the second crawl. For all methods n was set to 4 400. The same set of queries is used across all crawls. We fixed the queries in order to investigate what effect the dynamic underlying collections may have on the obtained representation sets across the different strategies. Examining the impact of changing user information needs on the quality of representation sets is a potential direction for further research.

CU. Using the constant update policy n_c was 100, which resulted in one hundred documents being added to each collection at each time step.

QL. 1 000 queries were randomly selected from the ABC query log to seed the QL method. None of the 448 testing queries exist in this random set. CRCS [Shokouhi, 2007] was employed to rank collections in each crawl set. The number of new documents that should be added to each representation set, is calculated using the Eq. (3).

SS. For the size based policy, at each time step the latest sizes of the collections were estimated using the capture-history method [Shokouhi et al., 2006]. Then, Eq. (4) was used to determine the number of new documents that should be sampled from each collection when $n = 4\,400$.

Baselines. These three policies were compared against two baselines; (1) the naive baseline where the representation sets remain static i.e $\theta_C^t = \theta_C^0$ which consists of 100 doc-

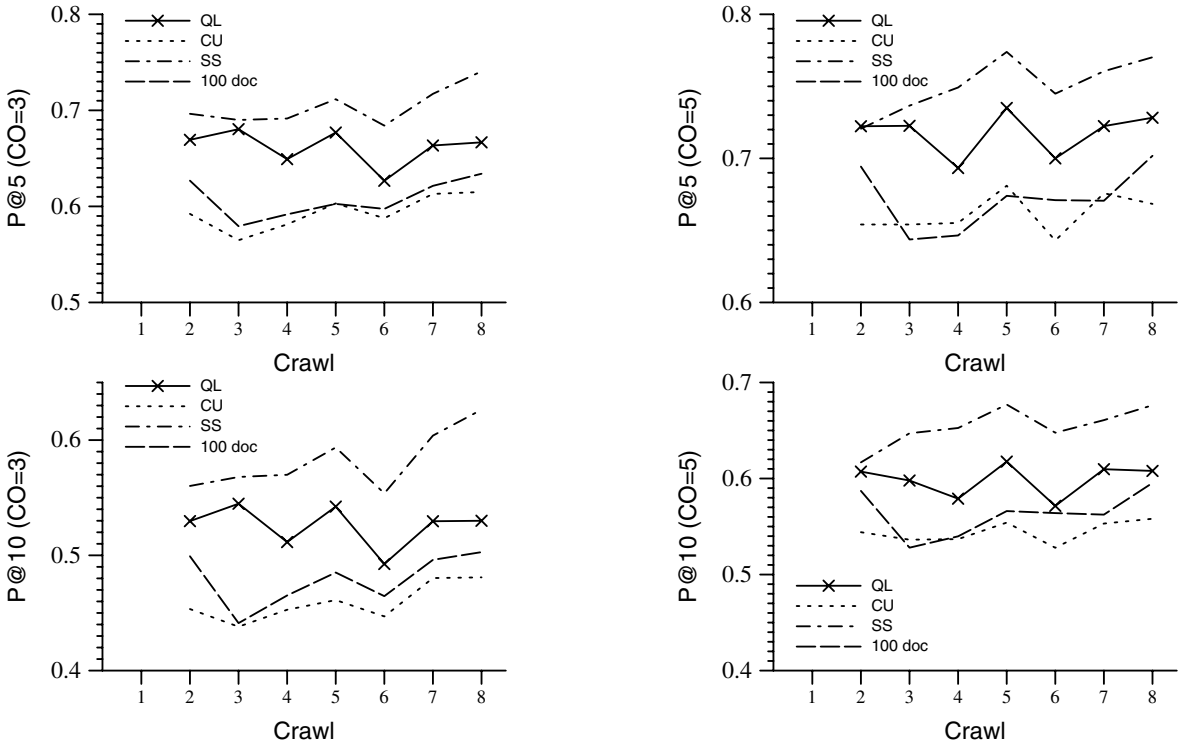

Figure 5: The P@5 and P@10 values produced by running 448 queries on different collection representation sets. The CO values represent the number of collections that are selected for a query.

uments sampled after the first crawl, and (2) the stronger baseline which creates a new representation set at each time step, by sampling 100 documents (and discarding the previous documents. i.e. $D_C^t = \{d_1^t, \ldots, d_{100}^t\}$).

Vocabulary of representation sets. Figure 4 compares the language models of representation sets with that of their corresponding collections in each crawl. As expected, the second baseline – using fresh representation sets at each crawl composed of 100 documents – remains consistently close to the actual collections. Conversely, by not updating the representation sets results in a gradual deterioration illustrated by the old-baseline. In comparison, the three updating policies display a consistent estimation close to the 100-doc baseline. This would indicate that by using an updating policy, the representation sets reflect the content changes for each collection. Note that QL, CU and SS do not evict the old sampled documents from collection representation sets. Some of these documents may be deleted or updated in collections over time. Therefore, the language models of representation sets for these methods are more *noisy* than that of the 100-doc baseline. Overall, there appeared to be minimal difference between updating policies in terms of the representation set vocabulary, except after the first crawl where the SS policy experienced a sharp increase in KL divergence. Across the subsequent crawls, the KL divergence for the SS policy began to converge towards the other polices.

One interesting observation during crawls seven and eight was that the KL divergence increased sharply for all three updating policies. This trend also corresponds to the large decrease in the number of documents in the testbed overall (see Table 1). This large deletion in content from a number of collections affected the corresponding representation sets, with those documents still contained in the representation set but not the collection increasing the KL divergence. This suggests that update policies should also consider how to deal with (remove) antiquated documents from the representations in order to be more accurate.

5.1 Retrieval performance

Adopting an updating policy improved performance in comparison to both baselines (Fig. 5). Overall, the SS policy provided the most accurate and consistent performance. When comparing the precision of SS against the 100-doc baseline, the policy was found to provide a significant improvement over time with the exception of crawl 2 ($p < 0.001$). This indicates that initially updating does not affect the performance, but over time the representations sets will gradually go out-of-date and not reflect the underlying collection content. It is not clear to what extent this improvement was a result of either the SS algorithm sampling more documents from the larger collections or because the larger collections are more dynamic in comparison to the smaller sized collections, or a combination of both factors. What it does indicate though is that a standard uniform sampling across collections is not an optimal strategy.

In comparison to SS, the QL and CU updating policies were not as consistent. QL recorded higher precision values on average than CU, although both approaches showed relatively worse performance than SS. When comparing QL against the baseline, the policy was found to significantly improve over the baseline during crawls 3 to 5 and crawl 7 ($p < 0.05$). Interestingly, using a naive policy, CU, resulted

in worse performance than the baseline. Although, the differences are not detected as being statistically significant.

6. CONCLUSIONS AND FUTURE WORK

We have argued and then illustrated through experimentation the utility of adopting an updating strategy for maintaining the representation sets of dynamically changing collections. If a representation set is not updated over time, the quality of the representation will deteriorate. This deterioration is a product of new vocabulary added to the collection, the removal of old vocabulary, and also the effect of bursts in vocabulary, where terms fluctuate sharply in both collection frequency and rank. If a representation set does not reflect these trends, then retrieval accuracy and consistency also deteriorates.

To address this problem we considered three policies for updating representation sets. Through experimentation, it was shown that updating larger collections more frequently (SS) is the most effective method and can significantly improve the retrieval performance. In comparison, out-of-date representations posed a significant problem to the operational effectiveness of a FIR system. Although updating did not always provide significant gains in the short term in comparison to the baseline (no updating), there was sufficient evidence to show that without updating retrieval accuracy becomes inconsistent and in the the long term both accuracy and reliability decline.

The SS policy updated the representation sets of the larger collections more than the smaller collections. This policy assumes that larger collections will have a larger proportion of updates. Also, larger collections tend to be more popular given the wider array of available content. However, further investigation would be required to confirm these assumptions as well as develop alternative methods based on individual collection characteristics.

Finally, obtaining representation sets for uncooperative collections in FIR using query-based sampling or focused probing shares parallels with hidden-web crawlers. A commonality shared amongst these techniques is that input queries are submitted to the search interface of a collection in order to access dynamically generated content [Ntoulas et al., 2005]. Therefore, new updating policies derived for hidden-web crawlers could also utilise features such as SS.

Acknowledgement. We are grateful to Halil Ali for providing the ABC dataset, and to David Elsweiler for his helpful comments.

References

Avrahami, T., Yau, L., Si, L., and Callan, J. (2006). The FedLemur: federated search in the real world. *Journal of the American Society for Information Science and Technology*, 57(3):347–358.

Baillie, M., Azzopardi, L., and Crestani, F. (2006). Adaptive query-based sampling of distributed collections. In *Proc. SPIRE Conf., Glasgow, UK*, pages 316–328.

Callan, J. (2000). *Advances in information retrieval*, Chapter 5, Distributed information retrieval, pages 127–150. Kluwer.

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

Callan, J. Lu, Z., and Croft, B. (1995). Searching distributed collections with inference networks. *Proc. ACM SIGIR Conf., Seattle, WA*, pages 21–28.

Cho, J. and Garcia-Molina, H. (2003). Effective page refresh policies for web crawlers. *ACM Transactions on Database Systems*, 28(4):390–426

Craswell, N., Bailey, P., and Hawking, D. (2000). Server selection on the World Wide Web. *Proc. ACM Conf. on Digital Libraries, San Antonio, TX*, pages 37–46,

Craswell, N., Crimmins, F., Hawking, D., and Moffat, A. (2004). Performance and cost tradeoffs in web search. In *Proc. Australasian Database Conf., Darlinghurst, Australia*, pages 161–169, Australian Computer Society, Inc.

Gravano, L., Chang, C., Garcia-Molina, H., and Paepcke, A. (1997). Starts: Stanford proposal for internet meta-searching. In *Proc. ACM SIGMOD Conf., Tucson, AZ*, pages 207–218.

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

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

Hawking, D. and Thomas, P. (2005). Server selection methods in hybrid portal search. In *Proc. ACM SIGIR Conf., Salvador, Brazil*, pages 75–82.

Ipeirotis, P., Ntoulas, A., Cho, J., and Gravano, L. (2005). Modeling and managing content changes in text databases. In *Proc. ICDE Conf., Tokyo, Japan*, pages 606–617.

Kleinberg, J. (2006). Temporal dynamics of on-line information systems. *Data Stream Management: Processing High-Speed Data Streams*.

S. Kullback. Information theoery and statistics. *Wiley, New York, NY*, 1959.

Ntoulas, A., Zerfos, P., and Cho, J. (2005). Downloading textual hidden web content through keyword queries. In *Proc. ACM/IEEE-CS Joint Conf. on Digital libraries, Denver, CO*, pages 100–109.

Paepcke, A., Brandriff, R., Janee, G., Larson, R., Ludaescher, B., Melnik, S., and Raghavan, S. (2000). Search middleware and the simple digital library interoperability protocol. *D-Lib Magazine*, 6(3).

Price, G. and Sherman, C. (2001). *The Invisible Web : Uncovering Information Sources Search Engines Can't See*. CyberAge Books.

Robertson, S., Walker, S., Hancock-Beaulieu, M., Gull, A., and Lau, M. (1992). Okapi at TREC. In *Proceedings of TREC-1992, Gaithersburg, MA*, pages 21–30.

Si, L. and Callan, J. (2003a). Relevant document distribution estimation method for resource selection. In *Proc. ACM SIGIR Conf., Toronto, Canada*, pages 298–305.

Si, L. and Callan, J. (2003b). A semisupervised learning method to merge search engine results. *ACM Transactions on Information Systems*, 21(4):457–491.

Si, L. and Callan, J. (2004). Unified utility maximization framework for resource selection. In *Proc. ACM CIKM Conf., Washington, DC*, pages 32–41.

Si, L., Jin, R., Callan, J., and Ogilvie, P. (2002). A language modeling framework for resource selection and results merging. In *Proc. ACM CIKM Conf., McLean, VA*, pages 391–397.

Shokouhi, M. (2007). Central-Rank-Based Collection Selection in uncooperative distributed information retrieval. *Proc. ECIR Conf., Rome, Italy*, pages 160–172.

Shokouhi, M., Zobel, J., Tahaghoghi, S., and Scholer, F. (2007). Using query logs to establish vocabularies in distributed information retrieval. *Journal of Information Processing and Management*, 43(1).

Shokouhi, M., Zobel, J., Scholer, F., and Tahaghoghi, S. (2006). Capturing collection size for distributed non-cooperative retrieval. In *Proc. ACM SIGIR Conf., Seattle, WA*, pages 316–323.

J. Xu and J. Callan (1998). Effective retrieval with distributed collections. In *Proc. ACM SIGIR Conf., Melbourne, Australia*, pages 112–120.

Xu, J. and Croft, W. B. (1999). Cluster-based language models for distributed retrieval. In *Proc. ACM SIGIR Conf., Berkeley, CA*, pages 254–261.