Federated Text Retrieval From Uncooperative Overlapped Collections

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

Justin Zobel
School of Computer Science and
Information Technology, RMIT University
Melbourne, Australia
jz@cs.rmit.edu.au

ABSTRACT

In federated text retrieval systems, the query is sent to multiple collections at the same time. The results returned by collections are gathered and ranked by a central *broker* that presents them to the user. It is usually assumed that the collections have little overlap. However, in practice collections may share many common documents as either exact or near duplicates, potentially leading to high numbers of duplicates in the final results. Considering the natural bandwidth restrictions and efficiency issues of federated search, sending queries to redundant collections leads to unnecessary costs.

We propose a novel method for estimating the rate of overlap among collections based on sampling. Then, using the estimated overlap statistics, we propose two collection selection methods that aim to maximize the number of unique relevant documents in the final results. We show experimentally that, although our estimates of overlap are not inexact, our suggested techniques can significantly improve the search effectiveness when collections overlap.

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

General Terms

Algorithms, Design, Experimentation

Keywords

Resource selection, federated search, distributed information retrieval, overlap estimation, overlapped collections

1. INTRODUCTION

In federated information retrieval (FIR), the query is sent simultaneously to several collections. Each collection evaluates the query and returns the results to the broker. As there is no need to directly access the index of the collections, FIR methods can search the hidden web. FIR can also provide a search service over the latest version of documents in collections without consuming costly resources for crawling and indexing.

For each query, the broker selects the collections that are most likely to return relevant documents. This creates the collection selection problem [Callan et al., 1995; Nottelmann and Fuhr, 2004; Hawking and Thomas, 2005; Si and Callan, 2003a]. To be able to select suitable collections, the broker acquires some knowledge about the contents of each collection, creating the collection representation problem [Baillie et al., 2006; Callan and Connell, 2001]. The knowledge of the broker about each collection may vary from detailed vocabulary statistics to a limited number of sampled documents. Once the selected collections return their results, the broker merges them and presents them to the user. This final step is the result merging problem [Callan et al., 1995; Si and Callan, 2003b].

Our paper focuses on the first problem: collection selection. FIR techniques assume that the degree of overlap among collections is either none or negligible [Si and Callan, 2003b]. However there are many collections such as bibliographic databases or news resources that have a significant degree of overlap. In such scenarios, selecting collections that are likely to return the same results not only wastes costly resources, but also degrades search effectiveness by introducing duplicate documents into the final results.

We propose a method that estimates the degree of overlap among collections by sampling from each collection using random queries. In addition, we introduce two collection selection techniques that use the estimated overlap statistics to maximize the number of unique relevant documents in the final results.

Our experiments on three testbeds suggest that, compared to the state-of-the-art methods, our techniques return fewer duplicate documents. They also significantly outperform the current alternatives in terms of the final search precision.

2. RELATED WORK

Many collection selection techniques are based on the assumption that the federated search environment is *cooperative*, that is, collections provide the broker with their index statistics and other useful information. CORI [Callan et al., 1995], GlOSS [Gravano et al., 1994], and CVV [Yuwono and Lee, 1997] are examples of such methods, among which CORI has been suggested to be the most effective

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...$5.00.

[Craswell et al., 2000; Powell and French, 2003], although there is also evidence that is poor [D'Souza et al., 2004].

In practice, federated search systems may be *uncooperative*. In this case, collections do not make their index statistics available to the broker. Therefore, the broker samples documents from each collection and uses them for collection selection. Recent collection selection algorithms were developed for uncooperative environments.

Si et al. [2002] proposed a language modelling framework for collection selection and result merging and showed that it can slightly outperform CORI in most cases. ReDDE [Si and Callan, 2003a] estimates the number of relevant documents in collections according to their sampled documents. Collections are ranked according to the number of their sampled documents that are ranked highly by a central model.

Nottelmann and Fuhr [2003] introduced a decision theoretic framework (DTF) for collection selection. It tries to minimize the overall costs of federated search including money, time, and retrieval quality. However, the effectiveness of DTF, in particular for short queries, has been found to be inferior to that of CORI [Nottelmann and Fuhr, 2003]. Nottelmann and Fuhr [2004] showed that combining DTF with CORI can increase its general effectiveness.

In a similar approach, Si and Callan [2004] proposed a unified utility maximization framework (UUM) for resource selection. As in ReDDE, UUM runs queries on an index of all sampled documents. It uses training queries to learn the probabilities of relevance for the sampled documents according to their central scores. Using these probabilities, UUM selects collections that are likely to maximize either the final precision or final recall.

RUM [Si and Callan, 2005] is an enhanced version of UUM that also considers the search effectiveness of resources. That is, collections are ranked according to the number of relevant documents that they are expected to return, instead of the number of relevant documents that they contain. Training queries are used to model the search effectiveness of collections. Si and Callan [2005] showed that the results produced by RUM are at least as good as those of UUM.

Hawking and Thomas [2005] suggested a hybrid approach that combines federated search with centralized techniques. In their method, the link anchor text available in a set of crawled documents is used to provide a description of collections that are not crawled. Collections are ranked according to the similarities of crawled pages referring to them. They showed that their technique can outperform ReDDE and CORI for some tasks.

Collection representation. The broker needs to gather descriptions of each collection before collection selection, . Collections are selected according to the similarities of their descriptions with the query. In cooperative environments, collections provide the broker with their own descriptions. In uncooperative environments, where collections communicate with the broker only in response to queries, a query-based sampling (QBS) technique [Callan and Connell, 2001] can be used to obtain collection descriptions. In QBS, *probe* queries are submitted to collections. The documents returned are downloaded and used as descriptions.

Several strategies have been suggested for selecting the probe queries. Callan and Connell [2001] chose the probe queries from a set of dictionary words or existing sample documents. Gravano et al. [2003] suggested that probe queries be selected from the nodes of a hierarchical classification tree. We found that selecting the probe queries from query logs can lead to better samples and significantly improves the search effectiveness [Shokouhi et al., 2007]. Hedley et al. [2004] selected the initial probe queries from the search interface of collections. Query-based sampling may be static or adaptive. In static QBS, a fixed number of documents (say 300) is downloaded from each collection [Callan and Connell, 2001]. In adaptive QBS [Baillie et al., 2006; Caverlee et al., 2006; Shokouhi et al., 2006a], sampling terminates according to the size of collections or the number of new terms in the most recent samples.

Result merging. Once the selected collections return their results, the broker merges them into a single list and presents them to the user. CORI [Callan et al., 1995] and SSL [Si and Callan, 2003b] are two well-known examples of such algorithms. We use SSL for our experiments, as it has found to be more effective than CORI [Si and Callan, 2003b].

Duplicate detection in FIR. Typically, federated search techniques assume that collections are independent and overlap-free. There are only two previous works that have explored the problem of overlapped collections.

We introduced a method for detecting duplicate and near-duplicate documents among the returned results [Bernstein et al., 2006]. In this approach, collections send a hash vector with each document they return to the broker. The broker detects the duplicate and near-duplicate documents by comparing the returned hash vectors and discards them during merging. The major drawback with this technique is that it may not be applicable for uncooperative environments, as it expects collections to use the same hash functions and to return a hash value per document.

COSCO is an overlap-aware collection selection method proposed by Hernandez and Kambhampati [2005] for uncooperative environments, which uses a large number of training queries to measure the degree of overlap among collections. For collection selection, each query is matched with the previous training queries. Then, according to the overlap statistics stored for the training queries, collections are selected in a way that is expected to minimize the number of duplicate documents in the final results. If the terms available in a new query do not exist in the previous training queries, COSCO cannot effectively estimate the overlap and select suitable collections. Also, Hernandez and Kambhampati [2005] test COSCO on a set of bibliographic datasets; its effectiveness for heterogeneous datasets and typical web pages has not been investigated.

3. OVERLAP ESTIMATION

Documents downloaded by query-based sampling are the only source of information about collections in uncooperative environments, so it is necessary to extract as much information as possible from the samples. Considering the efficiency restrictions and bandwidth limits, it is advantageous if the extra information is extracted from documents that are already downloaded from collections. Methods for estimating the size of collections such as sample-resample [Si and Callan, 2003a] and capture-history [Shokouhi et al., 2006b] already have such characteristics. They can estimate the size of collections with a small number of probe queries and sample documents. In this section, we introduce a novel method for estimating the degree of overlap among collections. Our technique uses the documents downloaded by query-based sampling for estimating the rate of overlap and does not require any additional information.

Suppose that we have two collections C_1 and C_2, and there are K overlap documents between them. We gather one sample from each collection using query-based sampling, S_1 from C_1 and S_2 is from C_2. Let m_1 and m_2 represent the documents from S_1 and S_2 that are in K. That is, m_1 and m_2 are the subsets of sampled documents that are selected from the overlapped documents between C_1 and C_2:

$$m_1 = S_1 \cap K \qquad \textit{and} \qquad m_2 = S_2 \cap K$$

Assuming that the samples are random, we can estimate the sizes of m_1 and m_2 as follows:

$$|m_x| = \frac{|S_x| \times K}{|C_x|} \quad \textit{where} \quad x \in \{1, 2\}$$

Here, $|C_x|$ is the size of collection C_x that can be estimated by the capture-history technique [Shokouhi et al., 2006b]. From another perspective, m_1 and m_2 can be regarded as two random samples from the population of overlapped documents. The probability of any given document from m_1 to be available in m_2 is $\beta = \frac{|m_2|}{K}$. Therefore, the probability of any given document from m_1 not to be available in m_2 is calculated as $1 - \beta$. The expected number of documents in m_1 that are available in m_2 can be calculated as below:

$$E(D) = \sum_{i=0}^{|m_1|} iP(i)$$

where the possible number of overlap documents is $0 < i < |m_1|$ and $P(i)$ is the probability of having exactly i documents in m_1 that are also available in m_2. (Note that by definition $E(D)$ is the expected number of documents in m_2 that are available in m_1.) Since $P(i)$ follows a binomial distribution, for the expected value of D we have:

$$E(D) = \sum_{i=0}^{|m_1|} i \binom{|m_1|}{i} \beta^i (1-\beta)^{|m_1|-i}$$

That is:

$$E(D) = \sum_{i=1}^{|m_1|} \frac{i \times |m_1|!}{i! \times (|m_1| - i)!} \beta^i (1-\beta)^{|m_1|-i}$$

$$= \alpha \sum_{i=1}^{|m_1|} \frac{(|m_1| - 1)!}{(i-1)! \times (|m_1| - i)!} \beta^{i-1} (1-\beta)^{(|m_1|-1)-(i-1)}$$

where $\alpha = |m_1| \cdot \beta$. By substituting $i - 1$ with another variable j we have:

$$= \alpha \sum_{j=0}^{|m_1|-1} \frac{(|m_1| - 1)!}{(j)! \times ((|m_1| - 1) - j)!} \beta^{j} (1-\beta)^{(|m_1|-1)-(j)}$$

According to the binomial theorem:

$$(x + y)^n = \sum_{l=0}^{n} \binom{n}{l} x^n y^{n-l}$$

which gives:

$$E(D) = \alpha \cdot (\beta + 1 - \beta)^{|m_1|-1} \tag{1}$$

Thus:

$$E(D) = \alpha = |m_1| \cdot \beta = \frac{|m_1||m_2|}{K} \tag{2}$$

Equation (2) shows the expected number of documents in m_1 that are common with m_2. Similarly, $E(D)$ is the expected number of documents in m_2 that are common with m_1. Thus, the number of overlap documents is independent of the collection sizes. Having the number of duplicate documents (D) within two samples it is possible to estimate the value of K as:

$$\hat{K} = \frac{|m_1||m_2|}{D} = \frac{|C_1||C_2| \times D}{|S_1||S_2|} \tag{3}$$

Once the number of duplicate documents among collections is estimated, it can be used in collection selection algorithms for maximizing the number of unique relevant documents in the final results. In the following sections, we introduce two methods that use the estimated overlap statistics for collection selection.

Algorithm 1 RELAX resource selection

```
RELAX-SELECTION($(G, w, s)$)
  INITIALIZE-GRAPH($(G, S)$)
$S \leftarrow \emptyset$
$Q \leftarrow V[G]$
while $Q \neq \emptyset$ do
  $u \leftarrow$ EXTRACT-MAX($(Q)$)
  $S \leftarrow S \cup \{u\}$
  for each vertex $v \in Adj[u]$ do
    $d[v] \leftarrow d[u] - w_e(u, w)$ //Relaxing
  end for
end while
```

4. THE 'RELAX' SELECTION METHOD

A federated search environment in the presence of overlapped documents among collections can be represented by a graph G. In this graph, each vertex is a collection and each edge indicates the existence of overlap documents between a pair of collections.

We aim to minimize the number of duplicate documents in the final merged list. For this purpose, the selection algorithm has to avoid simultaneously selecting collections that are likely to return the same answers.

In our RELAX collection selection technique, initially the number of relevant documents in each collection (vertex) is estimated. As in ReDDE [Si and Callan, 2003a] and UUM [Si and Callan, 2004], we rank all the sampled documents from collections for each query. Assuming that the top λ documents returned in this central ranking are relevant (we used $\lambda = 150$), the number of relevant sampled documents for each collection is counted. Then according to the estimated size of collection and the number of sampled documents, the total number of relevant documents in collection C is computed as $\hat{R} = \frac{r \times |C|}{|S|}$. Here, $\hat{R}$ is the estimated number of relevant documents in collection C and $|C|$ is the

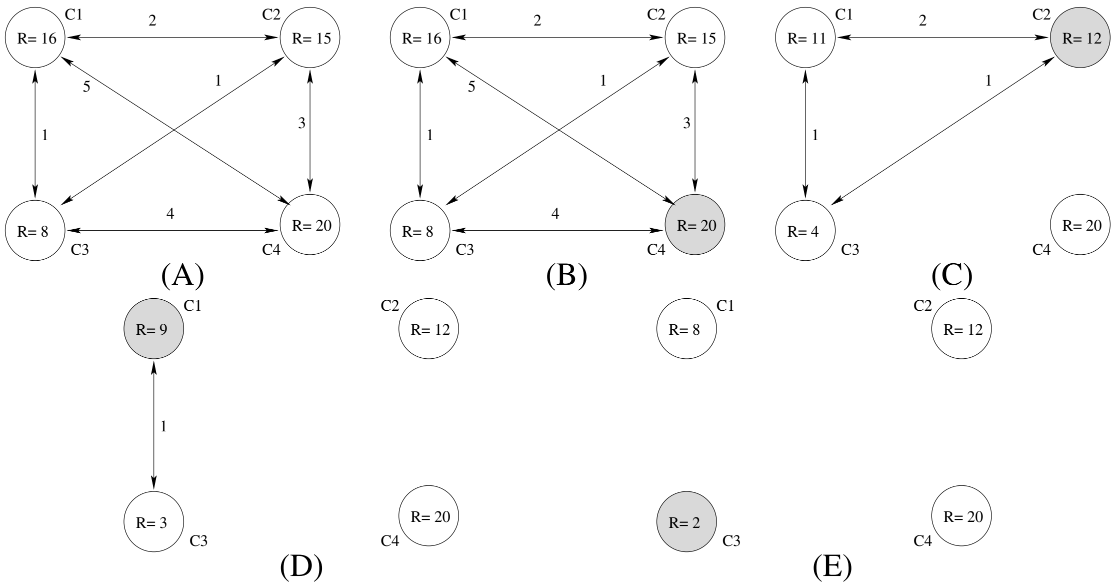

Figure 1: *The* Relax *selection on a sample graph. Each vertex (Cn) in this graph represents a federated collection. (A) The graph initialization where R represents the estimated number of relevant documents in each collection. (B) The graph after initialization where $C4$ is selected as the most relevant collection according to its R value. The weight $w_e(u, v)$ of an edge between u and v is computed according to the estimated number of documents common between u and v. (C)–(E) The status of the graph after selecting each collection (vertex).*

estimated size of collection. The number of sampled documents from collection C that are ranked in the top λ results by the central retrieval model is represented by r. Finally, $|S|$ is the number of sampled documents from collection C. Relax uses the estimated values for $\hat{R}$ as the weights of the collections.

In the next step, the weight $w_e(u, v)$ of an edge between two given collections u and v is calculated according to the approximated number of common relevant documents between u and v as follows:

$$w_e(u, v) = |\hat{R}_u \cap \hat{R}_v| = \frac{|\hat{R}_u \cup \hat{R}_v| \times \hat{K}}{|C_u \cup C_v|} \qquad (4)$$

Here, $|C_u \cup C_v|$ represents the total number of documents in both collections. $\hat{R}_u$ and $\hat{R}_v$ are respectively the estimated number of relevant documents in collections u and v. $\hat{K}$ is the estimated number of common documents between collections u and v that is calculated by Eq. (3).

At each stage, the collection with the highest number of relevant documents is selected. The weights of other collections are updated by subtracting the estimated number of their overlapped relevant documents with the selected collection (that is, by relaxing). In summary, our Relax selection method (Algorithm 1) is as follows:

1. Documents are downloaded from each collection using the query-based sampling technique.

2. The size of collections and the number of overlapped relevant documents between each pair of collections are estimated.

3. The federated environment is represented by a graph, where each vertex is a collection and the weight of each edge is computed using the number of common relevant documents between the connected pairs (Fig. 1).

4. The collection with the highest estimated number of relevant documents is selected.

5. Relax updates the graph by relaxing all collections and removing unnecessary edges.

6. Stop if there are no more edges or enough collections have been chosen. Otherwise, go to step 4.

Figure 1 shows a simple example of four overlapped collections. At the first step (A), the number of relevant documents in each collection R is estimated. At the next stage (B), the collection with the highest number of relevant documents ($C4$) is selected. The graph is relaxed by subtracting the estimated number of common relevant documents between the top collection and the connected collections. After each update, the edges connected to the most recent selected collection are removed. This process continues until there is no edge in the graph (C)–(E). That is, Relax selects collections according to the number of their unvisited relevant documents.

5. OVERLAP FILTERING FOR REDDE

Another strategy for avoiding duplicates in the final results is to remove collections with a high degree of overlap from the resource selection rankings. That is, initially the degree of overlap between collection pairs is estimated. Then for each query, collections are ranked using a resource selection method such as ReDDE [Si and Callan, 2003a]. Each collection at rank μ is compared with the other collections at the higher ranks. Collection C_μ is pruned from the original rank list if it has a large estimated overlap with at least

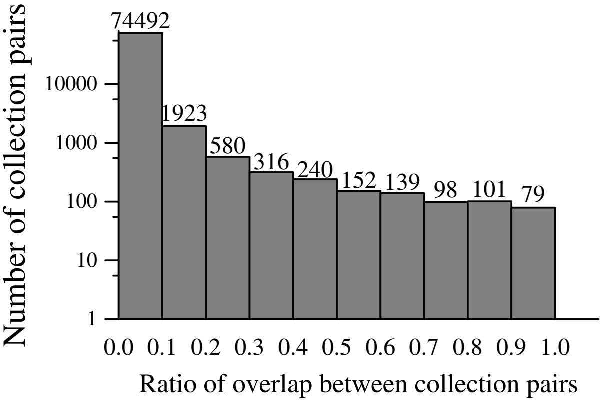

Figure 2: ***The overlap among collection pairs in the Qprobed-280 testbed.***

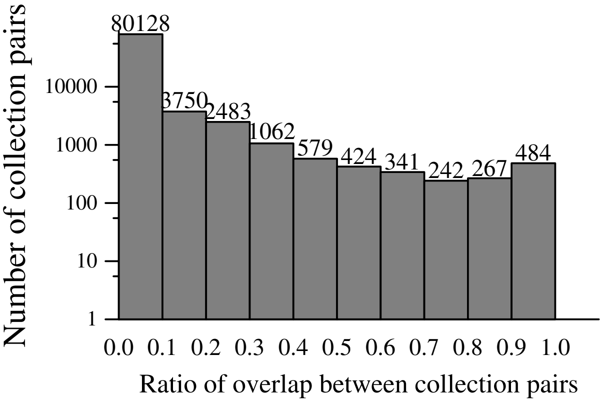

Figure 3: ***The overlap among collection pairs in the Qprobed-300 testbed.***

one of the other collections at higher ranks. Our filtering method for ReDDE, which we refer to as F-ReDDE, can be summarized as below:

1. The overlaps among collections are estimated as described for the Relax selection.
2. Collections are ranked using a resource selection algorithm such as ReDDE.
3. Each collection is compared with the previously selected collections. It is removed from the list if it has a high overlap (greater than γ) with any of the previously selected collections. We empirically choose $\gamma = 30\%$ and leave methods for finding the optimum value as future work.

The effectiveness of this method is expected to strongly depend on the underlying collection selection technique that is used. Also, the optimum value for γ may vary between testbeds. In the following sections, we compare the effectiveness of our selection methods with other techniques in the presence of overlap among collections.

6. TESTBEDS

The effectiveness of FIR methods tends to vary substantially on different testbeds [D'Souza et al., 2004; Si and Callan, 2003a]. Unfortunately, in the standard FIR testbeds [Powell and French, 2003; Si and Callan, 2003b], there is no overlap among collections. Thus, we create three new testbeds with overlapping collections based on the documents available in the TREC GOV dataset. We do not claim that our strategies for creating these testbeds are perfect or argue that the testbeds entirely reflect the characteristics of web collections. However, considering the available datasets for evaluating information retrieval experiments, we believe that the suggested testbeds are acceptable.

Qprobed-280. For this testbed, we used the 360 most frequent queries in in a query log provided by a major search engine, of queries with a highly ranked answer in the `.gov` domain. Each selected query is passed as a probe query to an index of TREC GOV documents. For each probe query, a random number of documents (between 5 000 and 20 000) are downloaded as a collection. Queries that return less than 5 000 answers are discarded.

In total, 280 collections with average size of 12 194 documents were generated. The largest and smallest collections in this testbed respectively contain 19 860 and 5 001 documents. Documents in each collection match for the same query and are likely to have somewhat similar topicality.

Figure 2 depicts the degree of overlap among collections in this testbed. There are 74 492 collection pairs that have less than 10% overlap while there are only 79 pairs with more than 90% of their documents in common. The overall rate of overlap among collections is low; only 1.1% of collection pairs in this testbed have more than 50% overlap.

Qprobed-300. Starting from the first collection in the previous testbed, every twentieth collection is merged into a single large collection. The same procedure is applied to every twentieth collection starting from the next initial 13 collections (collections $2, 3, ..., 14$) in the Qprobed-280 testbed. In total, the testbed is comprised of 300 collections. Figure 3 illustrates the degree of overlap among collection pairs. About 1.9% of collection pairs have more than 50% overlap which is higher than the Qprobed-280 testbed.

The collections in this testbed vary in size from 5001 to 180 985 documents with an average of 20 908 documents.

Sliding-115. We used a sliding window of 30 000 documents on the TREC GOV documents to generate 112 collections. Each collection has a random percentage of overlap $P\%$ $(25 < P < 100)$ with the previous collection. Then, starting from the first collection, every tenth collection is collapsed into a single large collection. The same procedure is applied on every tenth collection starting from the second and third collections, forming two additional large collections. In total, there are 115 collections. Figure 4 illustrates the degree of overlap among collection pairs on this testbed. About 2.5% of collection pairs have more than 50% overlap.

We expect that the impact of using overlap statistics becomes more significant as the overall overlap in the testbeds increases. The experimental results reported in the following sections confirm this hypothesis.

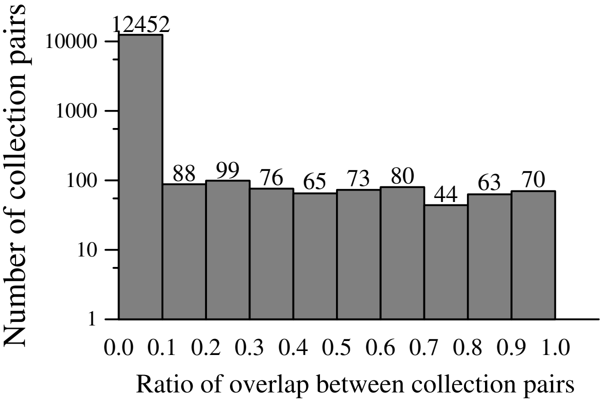

Figure 4: ***The overlap among collection pairs in the Sliding-115 testbed.***

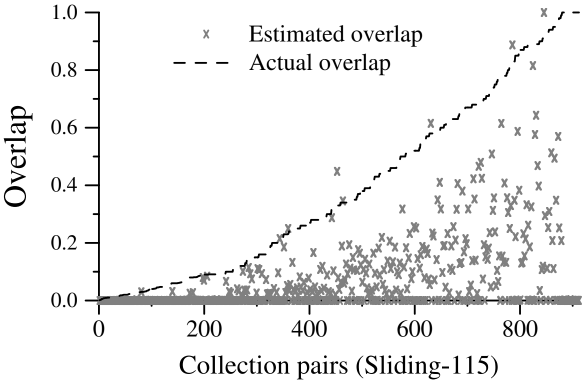

Figure 5: ***The accuracy of overlap estimation for collection pairs in the Sliding-115 testbed.***

7. RESULTS

The accuracy of our method for estimating the rate of overlap among collection pairs is measured using an *average estimation error* metric, defined as below:

$$AEE = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \frac{|D(i,j) - \hat{D}(i,j)|}{D(i,j)}$$

where

$$D(i,j) = \frac{|C_i \cap C_j|}{|C_i|} \tag{5}$$

Here, n is the total number of collections, $D(i,j)$ is the proportion of documents in collection i that are common with collection j. The value $|C_i \cap C_j|$ is equivalent to the K value in Eq. (3) and $|C_i|$ represents the size of collection i.

In our preliminary experiments, the initial estimated values for $D(i,j)$ suggested that the degree of overlap among collections is usually overestimated. This observation can be easily explained; document retrieval models are biased towards returning some popular documents for many queries [Garcia et al., 2004]. In addition, we found evidence that samples produced by query-based sampling are not random [Shokouhi et al., 2006b]. Therefore, the number of common documents between collection samples is often higher than the random scenario causing the overestimation of overlap in Eq. (3). Thus, we divide the estimated overlaps by the largest overlap value for normalization. In the rest of this paper and for all of our experiments, we use the normalized overlap values.

The AEE values computed for collections in the Qprobed-300, Qprobed-280, and Sliding-115 testbeds are respectively 0.91, 0.93, and 0.70.

Figure 5 shows the accuracy of estimations for overlapped collections in the Sliding-115 testbed. The horizontal axis represents the collection pairs sorted according to their actual overlap degree. It can be seen that the estimated values and the actual overlap rates correlate. The errors in estimations are largely due to two factors: (1) the query-based sampling technique does not produce random samples from collections [Shokouhi et al., 2006b] and (2) the size of samples are so small, so that they do not capture any duplicate document for estimating the degree of overlap. The trends for the other two testbeds were similar and we do not present them here.

In the rest of this section, we show that although our estimates of overlap are not perfectly accurate, our suggested methods can significantly improve the search effectiveness in the presence of overlap among collections.

Search effectiveness. To make our results comparable with previous work [Nottelmann and Fuhr, 2003; 2004; Si and Callan, 2003a;b; 2004; 2005], we run traditional static query-based sampling on each collection. Probe queries are selected randomly from the set of sampled documents and sampling terminates once 300 documents are downloaded. ReDDE is one of the most successful collection selection techniques that does not require training data. Therefore, we use it as the baseline of our experiments. We also compare the results with CORI.

For simplicity, we assume that all collections use the INQUERY [Callan et al., 1992] retrieval model and return at most 100 answers per query. We apply SSL [Si and Callan, 2003b]—the best current FIR merging method—to merge the results and compared methods by their precision values at early recall points ($P@n$). The statistical significant detected by the t-test for the differences between ReDDE and other methods at the 90% and 95% confidence intervals are respectively specified by $*$ and †. The duplicate documents in the final merged lists are considered as irrelevant.

Table 1 shows the precision values obtained by running the TREC topics 551–600 on the Qprobed-280 testbed. The cutoff values represent the number of collections selected per query. For the cutoff values 5 and 10, there is little difference between the effectiveness of methods, and only for P@5, Relax has a small advantage over the alternatives. When 20 collections are selected, the gaps become more apparent. Relax significantly outperforms ReDDE for precision at 5 and 10. F-ReDDE is at least as good as ReDDE and CORI produces a better P@5 value than ReDDE.

Table 2 includes the precision values produced by the selection methods on the Qprobed-300 testbed. The differences between ReDDE and Relax are often significant. Other methods have no significant advantage against each other, but ReDDE is usually the best among them.

On the Sliding-115 testbed, Relax produces the best results. It significantly outperforms ReDDE in 5 of 12 cases.

Table 1: Precision values obtained by collection selection methods on the *Qprobed-280* testbed. TREC topics 551–600 are used as queries. Cutoff values show the number of collections selected per query. For each query, the duplicate documents in the final merged lists are considered as irrelevant.

	Cutoff=5				Cutoff=10				Cutoff=20			
	P@5	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**
CORI	0.163	0.149	0.134	0.113	0.167	0.142	0.121	0.110	0.195^*	0.140	0.125	0.108
F-ReDDE	0.167	0.132	0.126	0.109	0.167	0.144	0.119	0.104	0.171	0.142	0.123	0.112
ReDDE	0.167	0.132	0.126	0.109	0.167	0.144	0.119	0.104	0.163	0.142	0.122	0.112
Relax	0.187^*	0.142	0.114	0.107	0.183	0.161	0.126	0.112	$0.208^\dagger$	$0.161^\dagger$	0.122	0.108

Table 2: Precision values obtained by collection selection methods on the *Qprobed-300* testbed. TREC topics 551–600 are used as queries. Cutoff values show the number of collections selected per query. For each query, the duplicate documents in the final merged lists are considered as irrelevant.

	Cutoff=5				Cutoff=10				Cutoff=20			
	P@5	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**
CORI	0.163	0.142	0.125	0.115	0.171	0.118	0.106	0.106	0.187	0.128	0.102	0.098
F-ReDDE	0.171	0.146	0.134	0.120	0.167	0.149	0.125	0.105	0.167	0.128	0.108	0.091
ReDDE	0.171	0.149	0.136	0.120	0.163	0.144	0.126	0.103	0.167	0.132	0.111	0.093
Relax	0.204^*	$0.177^\dagger$	0.140	0.128	0.187^*	0.153	0.130	$0.121^\dagger$	0.179	$0.157^\dagger$	$0.126^\dagger$	$0.111^\dagger$

Table 4: The average number of duplicate documents returned by each method per query for the *Qprobed-280* testbed. Cutoff values (CO) represent the number of collections selected.

	CO=5	CO=10	CO=20
CORI	20.58	27.66	35.94
F-ReDDE	15.22	25.95	33.04
ReDDE	15.22	26.57	34.49
Relax	13.51	22.57	32.04

Table 5: The average number of duplicate documents returned by each method per query for the *Qprobed-300* testbed. Cutoff values (CO) represent the number of collections selected.

	CO=5	CO=10	CO=20
CORI	20.28	29.73	42.20
F-ReDDE	33.20	49.95	55.32
ReDDE	32.81	50.69	56.00
Relax	30.51	39.51	48.08

F-ReDDE also produces better results than ReDDE in general. However, the differences are smaller and only significant at two cases. The precision values for CORI for cutoff=5 are specified in italic to indicate that they are significantly worse than ReDDE. This is consistent with observations of Si and Callan [2003a] suggesting that CORI is poorer than ReDDE on testbeds with skewed distributions of collection sizes. The differences between CORI and ReDDE become negligible for larger cutoff points.

As the extent of overlap among collections in the testbeds increases, the impact of using an overlap-aware collection selection method becomes more apparent. While there is no significant difference between methods on the Qprobed-280 testbed, the overlap-aware methods outperform the FIR baselines on the other two testbeds. This confirms our hypothesis that, as the overlap grows, there is a more noticeable need for use of overlap-aware selection methods.

Table 6: The average number of duplicate documents returned by each method per query for the *Sliding-115* testbed. Cutoff values (CO) represent the number of collections selected.

	CO=5	CO=10	CO=20
CORI	8.10	11.29	20.39
F-ReDDE	15.95	19.12	24.00
ReDDE	18.50	20.50	28.02
Relax	16.22	20.68	22.77

Number of duplicates. For each cutoff value, we compare the average number of duplicate documents returned by methods per query. Table 4 suggests that when the rate of overlap among collections is low, the number of duplicate documents returned by CORI, F-ReDDE, and ReDDE are usually comparable. Relax performs better than ReDDE and manages to reduce the number of duplicate documents by 11% and 15% respectively for cutoff values 5 and 10.

Tables 5 and 6 suggest that, compared to ReDDE, the overlap-aware selection methods can significantly reduce the chance of visiting a duplicate document in the final results. In both testbeds, CORI returns the lowest number of duplicate documents. This is due to the poor performance of CORI for testbeds with skewed distribution of collection sizes. Compared to the other methods, CORI selects the three large collections in the Sliding-115 testbed for fewer queries. The same observation can be made for the 14 large collections of the Qprobed-300col testbed. As these collections cause the highest overlap in the testbeds, missing them during collection selection significantly reduces the number of duplicate documents in the final results. However, Tables 2 and 3 show that the effectiveness of CORI on these testbeds is poor even when the number of duplicate documents is low. This is mainly because the large collections missing by CORI have a high density of relevant documents.

The average number of duplicates returned by ReDDE and F-ReDDE are similar on the Qprobed-300 testbed. On the Sliding-115 testbed, F-ReDDE returns 13% and 14% fewer duplicates respectively when 5 and 20 collections are

Table 3: The precision values obtained by collection selection methods on the *Sliding-115* testbed. TREC topics 551–600 are used as queries. Cutoff values show the number of collections selected per query. For each query, the duplicate documents in the final merged lists are considered as irrelevant.

	Cutoff=5				Cutoff=10				Cutoff=20			
	P@5	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**	**P@5**	**P@10**	**P@15**	**P@20**
CORI	*0.091*	*0.075*	*0.063*	*0.056*	0.187	0.139	0.108	0.095	0.179	0.145	0.140	0.128
F-ReDDE	0.166	0.143	0.122^*	0.107	0.170	0.141	0.125	0.110	0.166	0.141	0.137	$0.122^\dagger$
ReDDE	0.154	0.135	0.111	0.101	0.175	0.131	0.125	0.110	0.162	0.143	0.131	0.114
Relax	0.171^*	$0.154^\dagger$	0.122	0.115	0.187	0.156^*	0.133	0.115	0.154	0.152	$0.140^\dagger$	$0.133^\dagger$

selected. On the latter two testbeds, the number of duplicates returned by Relax is substantially less than ReDDE. Relax returns respectively 12% and 18% less documents than ReDDE for cutoff values 5 and 20 on the Sliding-115 testbed. On the Qprobed-300 testbed, the number of duplicates for Relax at CO=10 and CO=20 are respectively 22% and 14% less than that for ReDDE.

Overall, Relax produces the highest precision values. It also returns a lower number of duplicate documents than all methods but CORI. The F-ReDDE approach works well on some testbeds but on others is not significantly better than the alternatives. This might be due to the poor choice of γ, which was chosen based on our preliminary experiments.

8. CONCLUSIONS

We have introduced a novel technique for estimating the degree of overlap among uncooperative collections. Our method uses the sampled documents downloaded for collection selection and does not require any additional information. We have also proposed two overlap-aware collection selection techniques that consider the overlap statistics of resources for collection selection. Our experimental results show that, in the presence of overlap, our techniques can significantly outperform previous collection selection methods in terms of search effectiveness. They also lead to a smaller number of duplicate documents in the final merged results.

Several open directions remain as our future work. For our experiments in this paper, we used static query-based sampling and downloaded 300 documents for each collection. It is interesting to investigate the impact of sample size and sampling strategies on the accuracy of overlap estimations and the final search effectiveness. Moreover, we arbitrarily set γ to 30% according to preliminary experiments, but our results suggest that the best value for γ varies on different testbeds. Finally, although in theory the methods discussed in this paper are applicable for avoiding near-duplicate documents, they have not been tested for such a scenario.

References

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

Bernstein, Y., Shokouhi, M., and Zobel, J. (2006). Compact features for detection of near-duplicates in distributed retrieval. In *Proc. SPIRE Conf.*, pages 110–121, Glasgow, UK.

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

Callan, J., Croft, W. B., and Harding, S. (1992). The INQUERY retrieval system. In *Proc. Int. Conf. on Database and Expert Systems Applications*, pages 78–83, Valencia, Spain.

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

Caverlee, J., Liu, L., and Bae, J. (2006). Distributed query sampling: a quality-conscious approach. In *Proc. ACM SIGIR Conf.*, pages 340–347, Seattle, Washington.

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

D'Souza, D., Zobel, J., and Thom, J. (2004). Is CORI effective for collection selection? an exploration of parameters, queries, and data. In *Proc. Australian Document Computing Symposium*, pages 41–46, Melbourne, Australia.

Garcia, S., Williams, H., and Cannane, A. (2004). Access-ordered indexes. In *Proc. Australasian Computer Science Conf.*, pages 7–14, Darlinghurst, Australia.

Gravano, L., Garcia-Molina, H., and Tomasic, A. (1994). The effectiveness of GlOSS for the text database discovery problem. In *Proc. ACM SIGMOD Conf.*, pages 126–137, Minneapolis, Minnesota.

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.*, pages 75–82, Salvador, Brazil.

Hedley, Y., Younas, M., James, A., and Sanderson, M. (2004). A two-phase sampling technique for information extraction from hidden web databases. In *Proc. ACM Workshop on Web information and data management*, pages 1–8, Washington, DC.

Hernandez, T. and Kambhampati, S. (2005). Improving text collection selection with coverage and overlap statistics. In *Int. Conf. on World Wide Web*, pages 1128–1129, Chiba, Japan.

Nottelmann, H. and Fuhr, N. (2003). Evaluating different methods of estimating retrieval quality for resource selection. In *Proc. ACM SIGIR Conf.*, pages 290–297, Toronto, Canada.

Nottelmann, H. and Fuhr, N. (2004). Combining CORI and the decision-theoretic approach for advanced resource selection. In *Proc. Euorpean Conf. on Information Retrieval*, pages 138–153, Sunderland, UK.

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

Shokouhi, M., Scholer, F., and Zobel, J. (2006a). Sample sizes for query probing in uncooperative distributed information retrieval. In *Proc. Asia Pacific Web Conf.*, pages 63–75, Harbin, China.

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

Shokouhi, M., Zobel, J., Tahagoghi, S., and Scholer, F. (2007). Using query logs to establish vocabularies in distributed information retrieval. *Information Processing and Management*, 43(1):169–180.

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

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.*, pages 32–41, Washington, DC.

Si, L. and Callan, J. (2005). Modeling search engine effectiveness for federated search. In *Proc. ACM SIGIR Conf.*, pages 83–90, Salvador, Brazil.

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.*, pages 391–397, McLean, Virginia.

Yuwono, B. and Lee, D. L. (1997). Server ranking for distributed text retrieval systems on the internet. In *Proc. Int. Conf. on Database Systems for Advanced Applications (DASFAA)*, pages 41–50, Melbourne, Australia.