Robust Result Merging Using Sample-Based Score Estimates

14

MILAD SHOKOUHI
RMIT University
and
JUSTIN ZOBEL
RMIT University

In federated information retrieval, a query is routed to multiple collections and a single answer list is constructed by combining the results. Such metasearch provides a mechanism for locating documents on the hidden Web and, by use of sampling, can proceed even when the collections are uncooperative. However, the similarity scores for documents returned from different collections are not comparable, and, in uncooperative environments, document scores are unlikely to be reported. We introduce a new merging method for uncooperative environments, in which similarity scores for the sampled documents held for each collection are used to estimate global scores for the documents returned per query. This method requires no assumptions about properties such as the retrieval models used. Using experiments on a wide range of collections, we show that in many cases our merging methods are significantly more effective than previous techniques.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*Collection*

General Terms: Algorithms

Additional Key Words and Phrases: Result merging, result fusion, distributed information retrieval, uncooperative collections

ACM Reference Format:

Shokouhi, M. and Zobel, J. 2009. Robust result merging using sample-based score estimates. ACM Trans. Inform. Syst. 27, 3, Article 14 (May 2009), 29 pages. DOI = 10.1145/1508850.1508852 http://doi.acm.org/10.1145/1508850.1508852

This work was done when the authors were at RMIT University
Authors' addresses: M. Shokouhi, Microsoft Research, 7 JJ Thomson, CB3 0FB, Cambridge, U.K.; email: milads@microsoft.com; J. Zobel, NICTA, Level 2 Building 193, University of Melbourne, Parkville 3010, Australia; email: justin.zobel@nicta.com.au.

DOI 10.1145/1508850.1508852 http://doi.acm.org/10.1145/1508850.1508852

1. INTRODUCTION

Federated information retrieval (FIR) systems provide a single portal or *broker* to multiple search engines, or *collections*. Compared to the centralized systems, in which there is a single monolithic index of all documents, FIR systems can search the *hidden Web*, and can return the locally indexed Web pages without consuming costly resources for crawling. However, FIR systems are typically less effective than systems with centralized indexes, as they do not have access to the complete data and term statistics, and can potentially consume more resources at query time.

In FIR, each query is sent to several of the collections; the returned answers are then collated or *merged* into a single result list [Callan et al. 1995; Kirsch 2003; Si and Callan 2003c]. As a preliminary step, the broker must determine which subset of the collections each query must be sent to, via *collection selection* [Callan et al. 1995; Gravano et al. 1999; Fuhr 1999; Si and Callan 2003b, 2004, 2005].

In *cooperative* environments, collections provide broad information about their documents to the broker, which can use this information for collection selection and result merging. However, on the Web many collections are *uncooperative*. That is, they do not share their index statistics with the broker. In this scenario, the broker can sample a limited number of documents from each collection to approximate its corpus statistics [Callan and Connell 2001].

Three key problems must be solved to provide successful FIR: collection representation, collection selection, and result merging. In this article, we focus on the third problem—result merging (Figure 1). Current FIR result merging methods are either designed for cooperative environments [Callan et al. 1995; Kirsch 2003], or assume that collections return their document scores to the broker[Si and Callan 2003c]. As we discuss, these methods use document ranks or unreliable pseudoscores in the absence of document scores. In most practical situations where document scores are not available, these methods produce poor results.

We propose a novel approach to result merging for uncooperative environments, which does not require document scores to be published by collections. In our *sample-agglomerate fitting estimate* (SAFE) method, the query is run on the collection of sampled documents as well as broadcast to some of the original collections. The known scores on the samples, which are based on partial global statistics, are used to interpolate scores for the documents returned in response to the query from each collection. The success of the method is due to the fact that the scores for the sampled documents can provide fairly tight bounds and accurate estimates for the scores of the returned documents, even if there is no overlap, that is, none of the answers were in the sample.

Using experiments on a range of collections, we show that SAFE can outperform the principal alternatives. Though there are exceptions, in most cases SAFE is the better method.

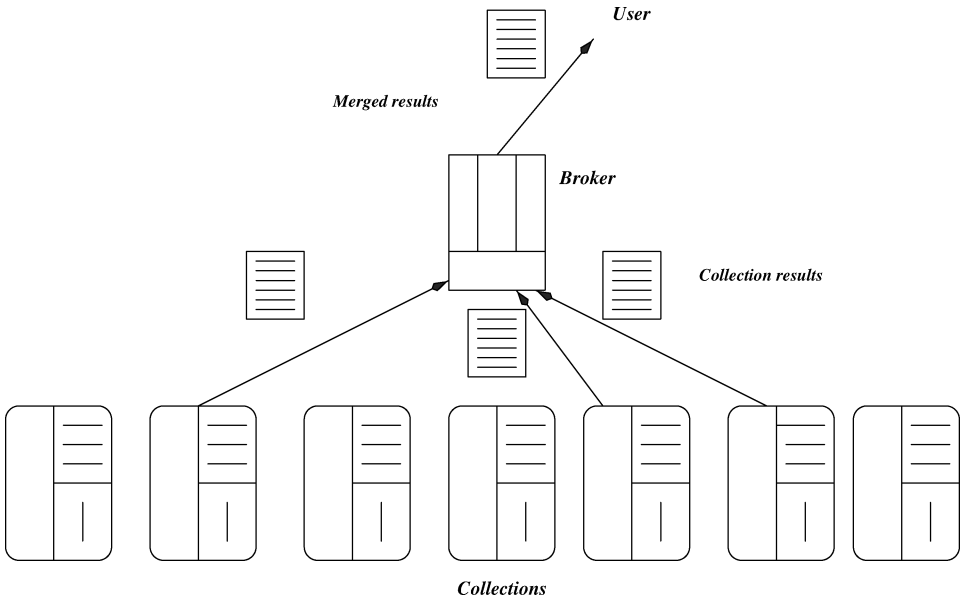

Fig. 1. Result merging process; selected collections return their top-ranked answers to the broker. The broker then merges those documents and returns them in a single list to the user.

2. RESULT MERGING

The goal of result merging algorithms is to calculate a global score for each document that is comparable to the scores of documents returned by other collections. Collections may use differing retrieval models and have different lexicon statistics. Thus, the document scores or ranks returned by multiple collections are not directly comparable and are not necessarily reliable for merging.

FIR merging, metasearch merging (collection fusion), and data fusion are similar but not identical concepts. In data fusion, different ranking functions are applied to the same collection [Aslam and Montague 2001; Aslam et al. 2003; Croft 2000; Fox and Shaw 1993; Lee 1997; Lillis et al. 2006; Oztekin et al. 2002; Wu and McClean 2006; Ng 1998; Vogt and Cottrell 1999; Vogt 1999]. In metasearch, the query is sent to multiple search engines [Dreilinger and Howe 1997; Glover et al. 1999; Lawrence and Giles 1998; Selberg and Etzioni 1995, 1997]. Assuming that the search engines have indexed the same collection (the Web), metasearch merge can be considered as a form of data fusion. Metasearch and FIR are not equivalent.

> Some authors define metasearch more broadly, as a unified search interface that queries multiple resources that may or may not overlap. This broader definition subsumes what we call metasearch and some aspects of [federated] information retrieval, although it usually assumes that all available resources are searched (i.e., no resource selection) [Si and Callan 2003c, page 459 (footnote)].

Moreover, current FIR algorithms use the sampled documents in collection representation sets for result merging. Metasearch engines, on the other hand, rely on the scores and the ranks of returned answers from search engines rather

than on sampled documents. We use the terminology as follows:

—In data fusion algorithms, different retrieval models are used on a single collection. Results returned by different models are merged to produce the final list. There is no form of collection selection or collection representation. The returned results from different models may wholly overlap as they are returned from the same collection.

—In metasearch merging, the results of different search engines for a query are merged. The broker may perform collection selection—usually according to the previous queries—or may simply send the queries to all collections. The results returned from different search engines may overlap although the indexes are substantially different. Usually, the advantage of metasearch is based on significant overlap. If the rate of overlap is low, effectiveness is also low [Wu and McClean 2007].

—In FIR merging, it is assumed that the rate of overlap among collections is either none or negligible. Collections are selected according to the similarities of their representation sets to the query. In uncooperative FIR environments, collection representation sets consist of sampled documents that are downloaded from collections and may also be used by merging algorithms.[1]

In this article, we focus on FIR merging.

2.1 FIR Merging

While collection selection and collection representation have seen relatively wide investigation in FIR, only a few approaches to result merging have been explored.

2.1.1 *CORI Merging.* In CORI merging [Callan et al. 1995], the global score D_G of a document returned by a collection (c) is computed based on its normalized document score (D') and collection score (C'). D' is the *collection-specific* weight of d that is returned by (c), and C' is the weight of c calculated by the broker.

$$C' = \frac{(C - C_{min})}{(C_{max} - C_{min})}, \tag{1}$$

$$D' = \frac{(D - D^c_{min})}{(D^c_{max} - D^c_{min})}, \tag{2}$$

$$D_G = \frac{D' + 0.4 \times D' \times C'}{1.4}. \tag{3}$$

C_{min} and C_{max} are, respectively, the minimum and maximum weights assigned to collections by the broker in the collection selection stage. D^c_{min} and D^c_{max} are

[1] We have recently investigated the performance of FIR methods on overlapped collections [Bernstein et al. 2006; Shokouhi and Zobel 2007; Shokouhi et al. 2007]. In this study however, we follow the assumption of disjoint collections to make our results comparable with previous work.

the minimum and maximum document scores reported by collection c. C and D are, respectively, the collection and document scores before normalization. Normalization parameters 0.4 and 1.4 have been suggested in the literature to keep document scores between zero and one [Callan 2000; Callan et al. 1995; Si and Callan 2003c]. CORI normalizes both collection and document scores. Larkey et al. [2000] showed that when both collection and document scores are normalized, the performance is better than scenarios in which only one of those scores is normalized.

2.1.2 *SSL Single-Model.* SSL [Si and Callan 2002, 2003c] is a semisupervised learning method that trains a regression model for each collection that maps document scores into their global scores. SSL creates a central index of all sample documents downloaded from collections. For a given query, some of the returned documents by collections may be already available in the central sample index. SSL compares the weights of such documents in the central index with the scores reported by collections to approximate the global scores of documents.

When collections use an identical retrieval model, SSL can use all of the overlap documents to train a single model that converts the collection-specific scores into global scores. In such a scenario—which we refer to as *SSL single-model*—for an overlap document $d_{i,j}$ returned from a selected collection c_i, SSL uses two scores: the score reported by the original collection ($D_{i,j}$) and the weight computed using the central sample-based index ($E_{i,j}$).

$$\begin{bmatrix} D_{1,1} & C_1 D_{1,1} \\ D_{1,2} & C_1 D_{1,2} \\ \cdots & \cdots\cdots \\ D_{n,m} & C_n D_{n,m} \end{bmatrix} \times [a \;\; b] = \begin{bmatrix} E_{1,1} \\ E_{1,2} \\ \cdots \\ E_{n,m} \end{bmatrix}. \tag{4}$$

Using the $D_{i,j}$ and $E_{i,j}$ values of all overlap documents, SSL trains a single regression model as below[2]:

$$D_G = a \times E_{i,j} + b \times E_{i,j} \times C_i, \tag{5}$$

where C_i is the weight of collection c_i that has returned document $d_{i,j}$. The combining parameters a and b can be estimated using a sufficient number of overlap documents. Si and Callan [2003c] suggested that at least three overlap documents are required for training the SSL models. More details about the SSL regression model can be found elsewhere [Si and Callan 2002, 2003c].

2.1.3 *SSL Multimodel.* When the retrieval models used in collections are not identical, SSL cannot train a single model that converts the outputs of all collections into global scores. The scores returned by collections may have different ranges. For example, KL-divergence language modeling [Lafferty and Zhai 2001] produces negative weights while INQUERY [Callan et al. 1997] produces positive weights between zero and one. Therefore, for each collection

[2]In a recent study, Paltoglou et al. [2007] have shown that, in the absence of document scores, using logistic functions instead of linear regressions may lead to slightly better results.

a separate model is trained that maps the collection scores to global values as below:

$$D_G = a_i \times E_{i,j} + b_i. \tag{6}$$

For a given document $d_{i,j}$ from collection c_i, D_G is the estimated global score and $E_{i,j}$ is the score of $d_{i,j}$ reported by collection c_i. The values for a_i and b_i can be obtained by training a regression matrix for each collection as follows:

$$\begin{bmatrix} D_{1,1} & 1 \\ D_{1,2} & 1 \\ \dots & 1 \\ D_{n,m} & 1 \end{bmatrix} \times [a_i \;\; b_i] = \begin{bmatrix} E_{1,1} \\ E_{1,2} \\ \dots \\ E_{n,m} \end{bmatrix}. \tag{7}$$

We refer to this technique as *SSL multimodel* in this article. Since a separate model is trained for each collection according to its returned answers, the likelihood of visiting an overlap document in the downloaded samples (training data) is lower than under SSL single-model. Therefore, the broker may need to receive longer result lists from collections or download some documents on the fly [Si and Callan 2003c]. Otherwise, SSL converts to CORI, which has been found to be a less effective method for merging [Si and Callan 2002, 2003c].

When document scores are absent, CORI and SSL assign *pseudoscores* to the returned answers [Si and Callan 2003c]. For example, when 1000 documents are returned from a collection, the score of the first-ranked document is set to 1, the next is set to 0.999, and so on. Rasolofo et al. [2003] also suggested the same strategy for computing the pseudoscores of documents when the scores are not available. However, the importance of answers might not be linearly comparable; typically, a few documents achieve high weight while most documents get negligible weight. As we show later, pseudoscores assigned in this way are not always effective. In addition, a user study [Joachims et al. 2005] suggested that, from the user's perspective, the importance of an answer does not have a linear correlation with its rank. A few top-ranked documents were found to be much more important than the others.

2.1.4 *Other FIR Merging Methods.* In the STARTS protocol [Gravano et al. 1997], collections return the term frequency, document frequency, term weight, and document weight information of each returned answer to the broker. Kirsch [2003] suggested that each collection should return the term frequencies, document frequencies, and the total number of indexed documents to the broker. In such methods, documents are merged according to their calculated similarities based on the received statistics by the broker.

As in CORI, Rasolofo et al. [2001] calculated the final score of a document by multiplying the document weight and collection score parameters. The document score in their approach is reported by its original collection. The collection score in their method is calculated according to the number of documents that are returned by each collection for the submitted query. This is based on a questionable assumption that collections returning a greater number of results for a query are more likely to contain relevant documents. Since their merging algorithm does not require collection statistics or representation sets, it is suitable

for both FIR and metasearch experiments. The same approach has been used by Abbaci et al. [2002] for merging.

Craswell et al. [1999] suggested that the broker can perform an effective merging by partially downloading the top returned documents (say the first 4 kB of each document) and using a reference index for the term statistics. They showed that the effectiveness of their approach is comparable to that of a merging scenario where documents are downloaded completely and the actual term statistics are used.

Xu and Croft [1999] applied a version of INQUERY [Callan et al. 1997] that uses the global inverse document frequency values to calculate the final score of each document for merging. The basic requirement for this approach is that collections provide the broker with the document frequency of each term in their index. This requires a significant exchange of information between collections and the broker, and is also only applicable to cooperative environments.

Wang and DeWitt [2004] used the PageRank of each returned answer for merging. In their approach, the final PageRank of a page d returned by a selected collection c is computed according to the estimated *ServerRank* of c and the computed *LocalRank* of d inside c. For calculating the values for d and c, the link information of all pages in collections is required.

2.1.5 *Summary.* In typical FIR experiments, it is usually assumed that collections do not overlap, and that snippets with the answers are not considered [Callan 2000; Callan and Connell 2001; Callan et al. 1995, 1999; Craswell et al. 2000; D'Souza and Thom 1999; D'Souza et al. 2004; French et al. 1999; Gravano and Garcia-Molina 1995; Gravano et al. 1994a, 1994b, 1999, 2003; Ipeirotis and Gravano 2004; Lu and Callan 2002; Nottelmann and Fuhr 2003; Ogilvie and Callan 2001; Paltoglou et al. 2007; Powell and French 2003; Rasolofo et al. 2001; Si et al. 2002; Si and Callan 2002, 2003a, 2003b, 2003c, 2004, 2005; Xu and Callan 1998; Xu and Croft 1999; Zobel 1997]. Current FIR merging algorithms make assumptions that are not always valid or realistic. For example, SSL assumes that collections return long results lists—say 1000 answers—for each query and there are overlap documents in the collection samples and the returned results. When overlap documents are not available, such methods assume that the broker downloads a few documents on the fly to obtain a sufficient number of overlaps. Our approach, described in the next section, is designed to address these shortcomings.

3. UNCOOPERATIVE RESULT MERGING

In uncooperative environments, each collection is assumed to return only a list of documents, without similarity scores or other such information. However, not only does the information available in samples taken from these collections allow estimation of those scores, but the estimates should be comparable between collections, allowing accurate result merging.

In uncooperative environments, collection summaries can be provided by the query-based sampling technique [Callan et al. 1999]. In query-based sampling, an initial query is selected—from a list of common frequent terms [Callan and

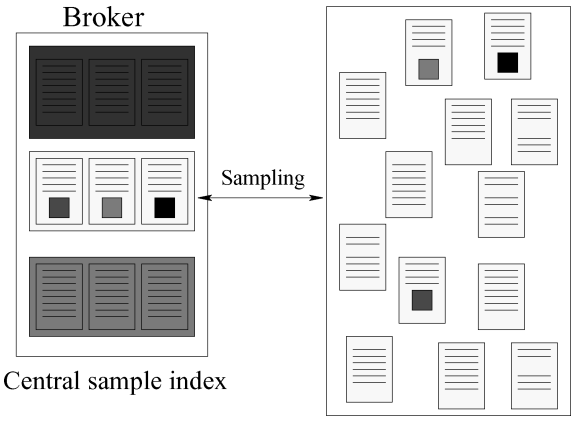

Fig. 2. A typical federated search system with three collections. Collection summaries are provided by sampling.

Connell 2001] or from the search interface of uncooperative collections [Hedley et al. 2004a, 2004b]—and is submitted to the collection. A few of the documents returned for the initial query are downloaded. The next query is selected from the text of the downloaded documents, and the process repeats. The sampling stops once a sufficient number of documents have been downloaded from each collection.

SSL [Si and Callan 2003c], which was outlined above, applies a semisupervised learning approach to estimate the scores of documents returned by collections. For each query, the weights of documents in collection samples are obtained by a similarity measure such as INQUERY [Callan et al. 1997]. If some of the documents returned by a collection are already sampled, their weights in the sample can be used to estimate the scores of other returned documents from that collection. The major drawback of SSL is that it cannot approximate the scores in the absence of overlap documents, a problem that becomes more acute when collections only return a few answers in their result lists and the likelihood of observing an overlap document is low. Thus, in an environment such as the Web, where typically only 10–20 answers are returned from collections, the SSL method requires downloading documents on the fly, or it backs off to less effective methods such as CORI [Callan et al. 1995].

Addressing such problems, we propose *SAFE* (sample-agglomerate fitting estimate), designed to work with the minimum cooperation between the broker and collections. SAFE uses the scores of all documents in the agglomeration of all the collection samples, and generates a statistical fit to estimate scores. It does not depend on the presence of overlap documents. SAFE is based on the following principle: For a given query, the results of the sampled documents is a subranking of the original collection, so curve fitting to the subranking can be used to estimate the original scores.

For example, consider the federated search system in Figure 2 with three collections. Collection summaries are generated by query-based sampling. Assume that θ_κ documents are sampled from collection c_κ (say κ = yellow). We run a query on these documents and apply a common similarity scheme such

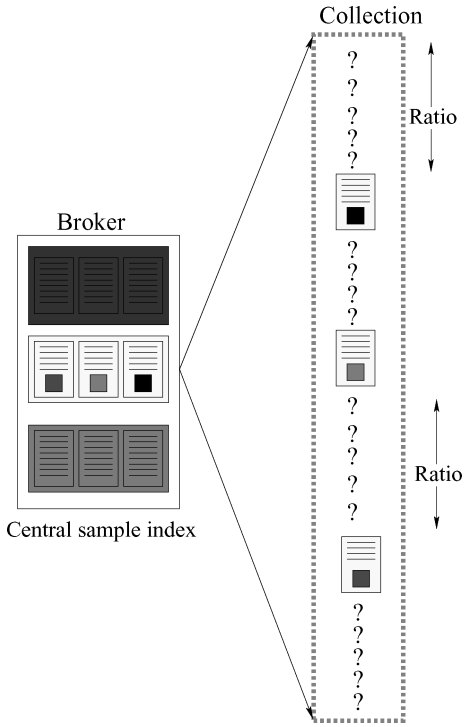

Fig. 3. Uniform distribution of sampled documents from collections.

as INQUERY [Callan et al. 1997] to calculate their similarity scores. We use the same similarity scheme to rank the documents in c_κ. If there is no overlap in the results returned by c_κ and the sampled documents, we can assume that the returned answers were ranked immediately ahead of all documents in the sample (we assume that all collections are running an effective weighting scheme, by which we mean a retrieval model that ranks documents in the order of their predicted relevance). As illustrated in Figure 3, we also assume that the documents in a sample—which is intended to be a random selection of documents from the collection—are uniformly distributed in the total ranking from that collection.[3] Therefore, the position (P) of a sampled document in the original collection results can be estimated as

$$P = r \times \mathit{Ratio}, \qquad \mathit{where} \qquad \mathit{Ratio} = \frac{|c_\kappa|}{|\theta_\kappa|}. \tag{8}$$

Here, r is the document rank when the query is executed on the sample, and $|c_\kappa|$ and $|\theta_\kappa|$ are, respectively, the number of documents in the collection c_κ and

[3]The documents downloaded by query-based sampling may not be a good random sample of collections [Shokouhi et al. 2006b; Thomas and Hawking 2007]. Therefore, the distribution of sampled documents is not likely to be uniform. This is due to different biases in the search engine retrieval models or the query sets [Bar-Yossef and Gurevich 2006; Bharat and Broder 1998; Garcia et al. 2004; Thomas and Hawking 2007]. However, we show later that, although the assumption of randomness is questionable, the accuracy of estimated scores is rather acceptable.

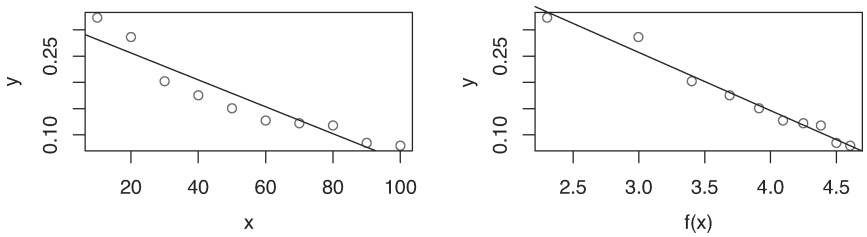

Fig. 4. Using a mapping function to improve the fitness of regression equations. The left graph shows the original data points. The y-axis represents the document scores and the x-axis denotes the document ranks. The graph on the right shows the same data points after using the mapping function $f()$. The mapped data points produce a fitter regression equation.

its sample.[4] In uncooperative situations, the collection sizes are usually unknown; SAFE estimates the size of each collection using the capture-history method [Shokouhi et al. 2006b]. In capture-history, a number of (say 140) random queries are sent to an uncooperative collection. The returned answers for each query are compared with the previously visited documents returned by the former queries. The size of collection is approximated based on the number of queries and their previously visited answers. Although the overhead of capture-history may not be small, to the best of our knowledge, it is currently one of the most efficient size estimation methods [Shokouhi et al. 2006b; Xu et al. 2007].

Having the weights of θ_κ data points—say $|\theta_\kappa| = 300$—of the original collection results, we can approximate the weights of other documents by curve fitting. In the presence of overlap documents, the P values are set according to the correct ranks indicated by the original collection. If there is no overlap in the results and the sampled documents, we assume that the returned answers were ranked immediately ahead of all documents in the sample (effective search in collections). For curve fitting, SAFE determines the relationship between the weights of sampled documents and their estimated ranks in collections by linear regressions:

$$w_d = m \cdot f(\hat{r}_d) + e, \tag{9}$$

where, w_d denotes the score of a sampled document d, and $\hat{r}_d$ is its estimated rank in the collection it has been sampled from. Parameters m (slope) and e (intercept) are constant variables, and $f()$ is a function for mapping the document ranks into different distributions. The mapping function changes the distribution of data points, in order to generate a fitter regression equation. The fitter the regression methods, the better they are for estimating the merging scores. Figure 4 shows the impact of using a mapping function on the sample data points. The y-axis represents the scores of documents sampled from a typical collection c. The x-axis on the left graph shows the estimated ranks of sampled documents in collection c. The right graph illustrates the same data points after

[4]A related estimation approach has been reported by Si and Callan [2004], but is applied to collection selection rather than result merging. Moreover, in their approach, documents must be downloaded from the collections to estimate the scores. Our SAFE method does not use any downloads and calculates the scores according to the similarities of documents in the sample.

Table I.

Name	$f(x)$	Model
LIN	$f(x) = x$	$w_d = m \cdot \hat{r}_d + e$
LOG	$f(x) = \ln(x)$	$w_d = m' \cdot \ln(\hat{r}_d) + e'$
SQRT	$f(x) = \sqrt{x}$	$w_d = m'' \cdot \sqrt{\hat{r}_d} + e''$
POW	$f(x) = \frac{1}{x}$	$w_d = m''' \cdot \frac{1}{\hat{r}_d} + e'''$
HYB	Hybrid model	

applying the mapping function $f()$ on the estimated ranks (x values). Using a mapping function has clearly improved the fitness of regression curve in this example. We show later that, if the mapping functions are chosen carefully, they often lead to fitter regression curves. We use the following five variants of $f()$, shown in Table I.

In experiments with the former four models, the merging scores are calculated based on the same regression model for all queries. Note that, in the LIN model, the mapping function does not change the initial score distribution. For the hybrid model (HYB), first, the goodness of curve-fitting for all models (LIN, LOG, SQRT, and POW) is computed according to their R^2 values [Gross 2003]. The model with the highest R^2 value is then used for calculating the final merging scores. Thus, the regression methods used for merging the results of different queries might not be identical.

3.1 Calculating the Merging Scores

The estimated scores for result returned from multiple collections are not comparable because the lexicon statistics of collection summaries are different. A major goal of result merging algorithms is, in effect, to calculate a global score for each returned answer so that it is comparable with the other results. The optimum ranking in FIR is expected to be similar to running the query on a monolithic index.

By making a few modifications to our estimation algorithm, we can use it to approximate the global scores of the collection results. The set of all samples together creates a central index that can be considered as a somewhat biased sample of the oracle global index. (The bias arises because the sample sizes may not be proportional to the sizes of the original collections, and also because query-based sampling does not produce random samples.) Therefore, the document scores in this index are representative of the weights in the global index.

Instead of running the queries on each collection sample individually, we can run the query on the aggregated index of all collection samples, and compute the w_d values in Equation (9) accordingly. We assume that the scores assigned by the central index to the sampled documents are representative of the global index scores. Using the estimation algorithm described above, we can approximate the scores of the returned documents, as if they were originally located in the central sample index. The scores calculated in this way are comparable because they are estimated using an identical retrieval model and according to the same lexicon statistics.

The only issue remaining is the number of data points required for training the regression equation. In situations where less than three documents are

ranked from a sample (the other documents do not contain the query terms), scores need to be approximated in another way. Joachims et al. [2005] reported the amount of time users spend looking at the document snippets in the search engine results. The curves reported by Joachims et al. [2005] follow the power law distribution. In the absence of sufficient data points, we use the distribution reported by them—as the eye fixation times—for different rank positions, and the weights of one or two available data points to approximate the global scores. However, it is unlikely that there will be fewer than three data points for a selected collection, as these collections are those whose sampled documents are found to be most similar to the query in the collection selection stage. Recent collection selection methods such as ReDDE [Si and Callan 2003b] and CRCS [Shokouhi 2007] rank collections according to the rankings of their sampled documents for queries. Hence, a collection selected by these methods is likely to have at least a few documents with nonzero scores.

In summary, the SAFE merging method is as follows:

—First, the central sample index ranks the sampled documents from all collections using an effective weighting scheme.

—Second, for each collection, the estimated size—obtained by the capture-history method [Shokouhi et al. 2006b]—is used to locate the position of the sampled documents in the original answer list.

—Finally, the global weights of documents from each collection are approximated by curve fitting.

4. TEST BEDS

We use six testbeds to evaluate the effectiveness of merging algorithms. These testbeds have been widely used in previous work [Callan 2000; French et al. 1999; Nottelmann and Fuhr 2003; Ogilvie and Callan 2001; Powell and French 2003; Si et al. 2002, Si and Callan 2003b, 2003c, 2005, Shokouhi 2007; Xu and Croft 1999].

—*trec123-100col-bysource (uniform)*. Documents in TREC disks one, two, and three are assigned to 100 collections by publication source and date [Powell and French 2003, Si and Callan 2003b, 2003c]. The <title> fields of TREC topics 51–100 are used as queries.

—*trec4-kmeans (trec4)*. A *k*-means clustering algorithm has been applied on the TREC4 data to allocate the documents into 100 homogeneous collections [Xu and Croft 1999]. The <description> fields of TREC topics 201–250 and their relevance judgments are used as queries.

—*trec-gov2-100col (gov2)*. First appearing in Shokouhi [2007], this is one of the largest testbeds available for federated search experiments. Documents in the TREC GOV2 dataset are partitioned into collections according to their first level of domain addresses. The testbed contains the largest 100 collections that are generated after partitioning. We used the <title> fields of TREC topics 701–750 for experiments on this testbed. Table II includes more information about the trec123-100col-bysource, trec4-kmeans and trec-gov2-100col datasets.

Table II. Testbed Statistics; trec123-100col-bysource Consists of 100 Collections Created from TREC Disks One, Two, and Three (Documents are assigned to collections according to their publication date or author. Trec4-kmeans is 100 collections generated from the TREC4 data. A clustering algorithm has been used to allocate documents to collections. Trec-gov2-100col is created from the largest 100 crawled servers in the TREC GOV2 dataset.)

	Size	No. of docs ×1000			Size (MB)		
Testbed	(GB)	Min.	Avg.	Max.	Min.	Avg.	Max.
trec123-100col-bysource	3.2	0.7	10.8	39.7	28	32	42
trec4-kmeans	2.0	0.3	5.7	82.7	4	20	249
trec-gov2-100col	110.0	32.6	155.0	717.3	105	1126	3891

—*trec123-AP-WSJ-60col (relevant)*. This and the remaining two testbeds are generated from the uniform testbed. In all of them, the <title> fields of TREC topics 51–100 and their relevance judgments have been used for retrieval evaluations. Documents in the 24 Associated Press and 16 *Wall Street Journal* collections in the uniform testbed are collapsed into two separate large collections. The other collections in the uniform testbed remain as before. The two largest collections in the testbed have a higher density of relevant documents for the TREC topics than do the other collections.

—*trec123-2ldb-60col (representative)*. Collections in the uniform testbed are sorted by their names. Every fifth collection starting with the first collection is merged into a large collection. Every fifth collection starting from the second collection is merged into another large collection. The other 60 collections in the uniform testbed are unchanged.

—*trec123-FR-DOE-81col (nonrelevant)*. Documents in the 13 *Federal Register* and 6 Department of Energy collections from the uniform testbed are merged into two separate large collections. The rest of collections remain unchanged. The two largest collections in the testbed have a lower density of relevant documents for the TREC topics than the other collections.

5. ACCURACY OF ESTIMATED SCORES

A perspective on the reliability of the scores estimated by SAFE is to compare them to the original documents scores. Differences between the correct document scores and their corresponding estimated values can be measured by the *mean squared error* (MSE). For a given collection c and a test query q, the MSE of estimated scores for the top n documents is calculated as

$$MSE(c, q, n) = \frac{1}{n} \sum_{i=1}^{n} (\widehat{w(c, q, d_i)} - w(c, q, d_i))^2. \tag{10}$$

Here, $w(c, q, d_i)$ is the score of the ith ranked document in collection c for the query q and $\widehat{w(c, q, d_i)}$ denotes the estimated score for the same document, calculated by SAFE as described in Section 3. The experimental setup for measuring the accuracy of score estimations is summarized below:

(1) For each collection c, we generate an index of its sample documents. For a test query q, the scores of sampled documents are computed using a document retrieval model M (we choose M to be INQUERY [Callan et al. 1997]).

Table III. The MSE Values Produced by SAFE Variations on Different Testbeds (Numbers are averaged over all collections and queries. TREC topics 301–400 are used as queries.)

	trec4	Uniform	Relevant	Nonrelevant	Representative	gov2
LIN	0.02	0.02	0.02	0.02	0.02	0.03
LOG	0.06	0.09	0.10	0.10	0.12	0.17
SQRT	0.03	0.04	0.04	0.04	0.04	0.06
POW	0.06	0.09	0.10	0.10	0.12	0.17

Table IV. The Goodness of Fit (R^2) for the Regression Models on Different Testbeds (The numbers are averaged over all collections and queries. TREC topics 301–400 are used as queries.)

	trec4	Uniform	Relevant	Nonrelevant	Representative	gov2
LIN	0.66	0.64	0.64	0.63	0.65	0.64
LOG	0.82	0.82	0.82	0.82	0.82	0.76
SQRT	0.79	0.77	0.76	0.77	0.77	0.71
POW	0.75	0.72	0.72	0.71	0.73	0.77
HYB	0.92	0.92	0.91	0.93	0.92	0.88

(2) We run q on c and use M to compute the scores of the top n documents.
(3) SAFE deploys the scores of sampled documents from c to estimate the scores of the top n documents returned in step 2. Note that the estimated scores here are not the same as the merging scores. Scores for merging are calculated based on the document weights in the central sample index.
(4) We use Equation (10) to compare the scores produced in steps 2 and 3.

Table III shows the accuracy of estimated scores for the top ten documents returned by collections on different testbeds. For each testbed, the MSE values are averaged over all queries (we used the <title> of TREC topics 301–400 for experiments in this section). For all methods, the average estimation errors are always less than 17% of the correct document scores. LIN and SQRT produce the smallest error rates. The MSE values for POW and LOG are between 4% to 14% worse than those produced by the former two models. We show later that smaller MSE values do not always lead to better search effectiveness, or to greater R^2 values.

LOG and POW perform noticeably worse than the other models on the gov2 testbed. Further investigations showed that their poor performance is due to the overestimation of document scores. For our experiments, the size of collection summaries is always 300 documents. Therefore, the *Ratio* factor in Equation(8) is greater for larger collections, in particular, for those in the gov2 testbed. This has a negative impact on the regression equations of LOG and POW, that are more sensitive to *Ratio* for estimating the scores of the top-ranked documents.

We also investigated the *goodness of fit* for different regression models by measuring their R^2 values [Gross 2003]. The results are presented in Table IV. For each testbed, the numbers are averaged over all queries and collections. SQRT and POW produce similar R^2 values, while LIN and LOG, respectively, have the lowest and highest fitness among the models. The low accuracy of LIN supports our earlier claim; if the mapping functions are selected with caution,

they can produce better-fitting curves than the LIN model. The hybrid approach produces the best R^2 values. It runs all the regression models for each query and selects the one with the maximum R^2 value for merging.

Overall, the results in Tables III and Table IV show that the scores estimated by SAFE (according to the sampled documents) are close to those actually reported by collections. They also suggest that the regression equations—for score estimation—are appropriate. Therefore, it is possible to estimate the scores of collection results even when there is little or no overlap between the collection answer lists and sampled documents.

6. EXPERIMENTS AND RESULTS

To measure the effectiveness of SAFE, we compare it to SSL on a range of scenarios. In the first set of experiments, collections use the same retrieval model (INQUERY [Callan et al. 1997]), while in the rest of experiments collections use various retrieval models. We assume that the environment is uncooperative and collections do not report the document scores. We used the Lemur toolkit[5] for our experiments. The collection indexes are stopped, and stemmed using the Porter stemmer [Porter 1997].

6.1 Parameters and Settings

There are many issues that need to be considered in comparing different FIR methods. Here, we explain parameters and settings used in our experiments.

6.1.1 *Result Lists (η).* For a given query, SSL uses documents that are returned by collections and are also available in the downloaded samples to calculate the global scores. The likelihood of visiting such documents has a direct relationship with the length of result lists. That is, the shorter the result lists are, the smaller is the chance of visiting such duplicate documents. We compare the performance of methods for both short and long result lists. For short lists, we assume that each collection returns at most ten documents per query ($\eta = 10$). This is the number that many commercial search engines such as Yahoo! and Google return for each query in their first page of results by default. For long lists, we assume that collections return at most one hundred answers for a query ($\eta = 100$). This is currently the maximum number of results that can be fetched from Yahoo! and Google for a query, using their advanced search functions. Extracting more answers usually requires visiting further pages of results and resubmitting the query.

6.1.2 *Cutoff Values.* Cutoff (CO) values show the number of collections that are selected for each query. Avrahami et al. [2006] suggested that selecting three to five collections is usually sufficient for extracting most of the available relevant documents. We use $CO = 3$ and $CO = 5$ for all experiments reported in this article.

6.1.3 *Evaluations.* The effectiveness of FIR is evaluated according to the number of relevant documents in the top-ranked merged results (P@n) [Si and

[5]http://www.lemurproject.org.

Callan 2003c]. We use precision at 5 and 10 for comparing the performance of merging algorithms. We use the bilateral t-test to measure the statistical significance of difference between the results of SAFE and SSL methods. Statistical significance at the 0.90, 0.95, and 0.99 confidence levels is specified by $*$, †, and ‡, respectively.

6.1.4 *Baselines.* The main contribution of this article is an FIR merging algorithm that is especially proposed for uncooperative environments. We compare our method with SSL as the state-of-the-art FIR merging algorithm. Si and Callan [2003c] suggested that when collections use an identical retrieval model, SSL single-model is the most appropriate approach. Therefore, in experiments with an identical retrieval model for all collections, we use SSL single-model as the baseline. For environments that collections use different retrieval models, Si and Callan [2003c] showed that SSL multimodel is a better option than SSL single-model. Thus, we use SSL multimodel as the baseline of experiments with multiple retrieval models.

6.2 Collection Selection

Once the collection samples are provided, SAFE can estimate the global scores of the returned answers. We use CRCS [Shokouhi 2007] for collection selection. Shokouhi [2007] showed that CRCS is more robust than some other state-of-the-art collection selection algorithms, such as ReDDE [Si and Callan 2003b] and CORI [Callan 2000]. A few collection selection algorithms such as UUM [Si and Callan 2004], and RUM [Si and Callan 2005] have been found to be more effective than ReDDE (and possibly than CRCS, although not tested by experiments). However, these methods require training queries that may not be available in practice.

As in ReDDE [Si and Callan 2003b], CRCS ranks collections according to their estimated number of relevant documents. However, unlike ReDDE that treats all the top-ranked documents equally, CRCS varies the importance (scores) of top-ranked documents according to their ranks. In CRCS, the broker creates a central index of all sampled documents. Each submitted query will be executed on this index before being sent to collections. The scores of collections are calculated according to their contributions to the top-ranked documents in the central sample index:

$$w_{c,q} = \frac{|\bar{c}|}{|\theta_c|} \times \sum_{d \in \theta_c} s(d, q, r), \tag{11}$$

where $w_{c,q}$ is the score of collection c for the query q, and $|\theta_c|$ is the number of sampled documents from c (always 300 in this article). $|\bar{c}|$ represents the size of collection c estimated by the capture-history method (the estimated sizes are divided by the size of the largest collection for normalization). $s(d, q, r)$ denotes the importance (score) of a sample document $d \in \theta_c$ in the central sample index at rank r, and is calculated as follows:

$$s(d, q, r) = \begin{cases} \alpha \exp(-\beta \times r), & \text{if } r < \gamma, \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

Table V. Precision Values for the Merging Methods on the trec4 Testbed (CRCS is used for collection selection. TREC topics 201–250 (long) and their relevance judgments have been used as queries. Collections use an identical retrieval model (INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	0.28	0.24	**0.33**	0.28	0.28	0.23	0.32	0.25
LIN	0.25	0.25	0.31	0.23	0.18	0.19	0.33	0.25
SQRT	0.27	0.26	0.29	0.27	0.18	0.21	0.34	0.31
LOG	0.27	0.27	0.30	0.27	0.17	0.21	0.34	0.30
POW	0.22	0.24	0.06	0.07	0.24	0.26	0.06	0.08
HYB	**0.34***	**0.30**‡	0.31	**0.29**	**0.32**†	**0.30**	**0.35**†	**0.32**†

We use $\gamma = 50$, $\alpha = 1.2$, and $\beta = 0.28$ in our experiments.[6] That is, the score of each collection is calculated according to its contribution to the top $\gamma = 50$ documents returned by the central sample index. However, the importance of the top γ documents varies exponentially according to their ranks. Therefore, sampled documents have different contributions to the final collection scores.

6.3 Merging with Single Retrieval Models

Our results in this section show the effectiveness of methods when collections use an identical retrieval model (INQUERY [Callan et al. 1997]). Document scores are not provided by collections in any of our experiments. To create collection samples, we download 300 documents from each collection by query-based sampling [Callan and Connell 2001]. Recent studies have suggested that using fixed-size samples for all collections is not always suitable, and adaptive sampling techniques are better to be used instead [Azzopardi et al. 2006; Baillie et al. 2006; Caverlee et al. 2006; Shokouhi et al. 2006a]. However, we use a fixed sample size for collections to make our results comparable with other related work.

The effectiveness of methods on the trec4 testbed is compared in Table V. The bold numbers represent the maximum precision values achieved by a method in each experiment. When three collections are selected, HYB consistently performs better than SSL. The improvements are up to 25% (for P@10), and are always statistically significant. For longer answer lists, both methods produce comparable results. For the cutoff value of five, HYB outperforms SSL at all levels. The differences are statistically significant ($p < 0.05$) in three of four cases. The less effective variants of SAFE are often dominated by SSL.[7] We found similar trends in all the other testbeds and hence, we only focus on

[6]In the original CRCS article [Shokouhi 2007], the value of β was mistakenly reported as 2.8 instead of 0.28.

[7]Note that for large values of η, the value of $1/\hat{r}_d$ in the POW regression model (Equation (9)) becomes very small. As a result, the document scores are significantly overestimated, and the accuracy of regression model decreases dramatically.

Table VI. Precision Values for the Merging Methods on the "Uniform" Testbed (CRCS is used for collection selection. TREC topics 51–100 (short) and their relevance judgments have been used as queries. Collections use an identical retrieval model (INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	0.33	0.34	**0.36**	**0.34**	0.33	0.34	0.34	0.32
LIN	0.31	0.31	0.29	0.29	0.30	0.31	0.30	0.28
SQRT	0.32	0.32	0.31	0.31	0.30	0.31	0.32	0.31
LOG	0.33	0.32	0.30	0.30	0.31	0.32	0.31	0.31
POW	0.30	0.31	0.10	0.10	0.31	0.32	0.14	0.13
HYB	**0.36**	0.34	0.34	0.30	**0.37**	**0.35**	**0.37**	**0.36**

Table VII. Precision Values for the Merging Methods on the "Relevant" Testbed (CRCS is used for collection selection. TREC topics 51–100 (short) and their relevance judgments have been used as queries. Collections use an identical retrieval model (INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	**0.31**	**0.28**	**0.32**	0.28	**0.31**	**0.28**	**0.34**	0.30
LIN	0.26	0.26	0.22	0.18	0.23	0.24	0.22	0.19
SQRT	0.24	0.24	0.21	0.19	0.22	0.23	0.22	0.20
LOG	0.24	0.24	0.21	0.20	0.23	0.22	0.22	0.21
POW	0.22	0.23	0.14	0.14	0.20	0.21	0.16	0.17
HYB	0.28	0.26	0.29	0.28	0.28	0.25	0.30	0.30

comparing HYB and SSL for the rest of this article. For the same reason, we only report the statistical significant tests for HYB and SSL.

Table VI reports the precision values obtained by the merging methods on the uniform testbed. As in the previous testbed, HYB, often outperforms SSL for $\eta = 10$ by up to 11% (for P@10, cutoff = 5). For long result lists, there is no noticeable advantage for one method against the other, and the precision values are similar.

The precision values produced by the merging methods on the relevant testbed are presented in Table VII. SSL generally outperforms HYB, and produces better results for both values of η. The gaps are usually between 9%–13%. However, the t-test does not detect any statistically significant difference.

On the nonrelevant testbed (Table VIII), there is no noticeable difference between HYB and SSL when three collections are selected, and $\eta = 10$. However, in all the other experiments, HYB performs substantially better than SSL. The improvements range between 5%–33%, and are statistically significant in many cases.

Table VIII. Precision Values for the Merging Methods on the "Nonrelevant" Testbed (CRCS is used for collection selection. TREC topics 51–100 (short) and their relevance judgments have been used as queries. Collections use an identical retrieval model (INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	0.34	**0.33**	0.32	0.31	0.34	0.33	0.33	0.31
LIN	0.33	0.29	0.32	0.28	0.32	0.32	0.34	0.30
SQRT	0.32	0.30	0.34	0.28	0.31	0.32	0.36	0.30
LOG	0.32	0.31	0.33	0.28	0.30	0.33	0.36	0.30
POW	0.28	0.28	0.13	0.11	0.32	0.31	0.15	0.13
HYB	0.34	0.32	**0.39***	**0.33**	**0.36**	**0.35**	**0.44**†	**0.37***

Table IX. Precision Values for the Merging Methods on the "Representative" Testbed (CRCS is used for collection selection. TREC topics 51–100 (short) and their relevance judgments have been used as queries. Collections use an identical retrieval model (INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	0.39	0.35	0.38	0.34	0.39	0.35	0.37	0.34
LIN	0.37	0.35	0.30	0.29	0.35	0.36	0.29	0.27
SQRT	0.38	0.36	0.30	0.28	0.34	0.36	0.30	0.28
LOG	0.38	0.37	0.31	0.28	0.35	0.37	0.30	0.28
POW	0.36	0.35	0.14	0.15	0.34	0.34	0.14	0.15
HYB	**0.43**	**0.36**	**0.39**	**0.36**	0.39	**0.37**	0.37	**0.36**

Tables IX and X, respectively, compare the merging methods on the representative and gov2 testbeds. The results do not show any statistically significant difference between SSL and HYB on any of these testbeds. HYB produces slightly better results on the representative testbed, while SSL performs somewhat better on gov2.

6.4 Merging with Multiple Retrieval Models

In the experiments reported so far, we assumed that collections are using the same retrieval model. To investigate the effectiveness of merging algorithms when collections use different retrieval models, we sequentially assign various models to collections. In all testbeds we sort collections by their names. We assign a variant of $tf \cdot idf$ [Zhai 2001] to every third collection starting from the first collection.[8] We apply KL-divergence language modelling [Lafferty and

[8]The $tf \cdot idf$ variant is based on the OKAPI formula derived from a probabilistic model as implemented in the Lemur toolkit [Zhai 2001].

Table X. Precision Values for the Merging Methods on the gov2 Testbed (CRCS is used for collection selection. TREC topics 701–750 (short) and their relevance judgments have been used as queries. Collections use an identical retrieval model (INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	**0.15**	**0.14**	**0.15**	**0.13**	**0.15**	**0.14**	0.17	0.13
LIN	0.11	0.10	0.13	0.12	0.11	0.10	0.13	0.10
SQRT	0.10	0.10	0.11	0.10	0.11	0.11	0.10	0.09
LOG	0.10	0.10	0.11	0.09	0.11	0.10	0.11	0.10
POW	0.11	0.12	0.02	0.02	0.13	0.14	0.08	0.06
HYB	0.13	0.12	0.14	0.12	0.14	0.13	0.17	**0.15**

Table XI. Precision Values for the Merging Methods on the trec4 Testbed (CRCS is used for collection selection. TREC topics 201–250 (long) and their relevance judgments have been used as queries. Collections use different retrieval models (tfidf, KL-Divergence, INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	0.19	0.20	**0.28**	**0.26**	0.20	0.20	**0.30**	**0.28**
LIN	0.18	0.20	0.20	0.18	0.12	0.15	0.22	0.19
SQRT	0.20	0.20	0.23	0.21	0.14	0.16	0.25	0.22
LOG	0.18	0.20	0.23	0.21	0.13	0.17	0.25	0.22
POW	0.21	0.19	0.08	0.08	0.22	0.22	0.09	0.10
HYB	**0.26**†	**0.24***	0.26	0.23	**0.24**‡	**0.24**	0.29	0.26

Zhai 2001] to every third collection starting from the second collection, and apply INQUERY [Callan et al. 1997] to the remaining collections.

Si and Callan [2003c] recommended that SSL multimodel should be used for situations that collections use different retrieval models. Therefore, we use SSL multimodel as the baseline of our experiments in this section. We are primarily interested in the experiments where 10 answers are returned per collection, as they are more similar to the real-world scenarios.

Table XI shows the effectiveness of merging methods on the trec4 testbed when different retrieval models are involved. As in the single retrieval model experiments, HYB dominates SSL consistently for short answer lists ($\eta = 10$). The differences are statistically significant in three of four cases. SSL outperforms HYB on long answer lists, but never produces—statistically—significantly better results.

The results in Table XII, do not show any major difference between HYB and SSL on the uniform testbed. This is consistent with the observations in Table VI, in which a single retrieval model was used across all collections. Except for one case, HYB consistently outperforms SSL, improving the precision values by up to 12%.

Table XII. Precision Values for the Merging Methods on the "Uniform" Testbed (CRCS is used for collection selection. TREC topics 51–100 (short) and their relevance judgments have been used as queries. Collections use different retrieval models (tfidf, KL-Divergence, INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	0.33	0.33	**0.37**	0.33	0.33	0.33	0.38	0.35
LIN	0.36	0.33	0.29	0.29	0.30	0.31	0.31	0.29
SQRT	0.33	0.33	0.31	0.30	0.29	0.31	0.33	0.31
LOG	0.34	0.32	0.30	0.29	0.30	0.30	0.33	0.31
POW	0.32	0.33	0.10	0.11	0.33	0.32	0.14	0.13
HYB	**0.37**	**0.34**	0.35	**0.34**	**0.35**	**0.34**	**0.39**	**0.37**

Table XIII. Precision Values for the Merging Methods on the "Relevant" Testbed (CRCS is used for collection selection. TREC topics 51–100 (short) and their relevance judgments have been used as queries. Collections use different retrieval models (tfidf, KL-Divergence, INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	**0.33**	**0.28***	0.27	0.23	**0.32**	**0.29**†	0.23	0.20
LIN	0.28	0.27	0.22	0.20	0.25	0.25	0.20	0.19
SQRT	0.26	0.24	0.22	0.20	0.23	0.23	0.20	0.20
LOG	0.28	0.24	0.21	0.20	0.23	0.21	0.20	0.19
POW	0.19	0.23	0.16	0.16	0.18	0.21	0.17	0.16
HYB	0.28	0.25	**0.30**	**0.29**	0.26	0.23	**0.29**	**0.28**

On the relevant testbed (Table XIII), SSL is the dominant method for short answer lists, and the differences are usually statistically significant for P@10 ($p < 0.1$ for cutoff $= 3$, and $p < 0.05$ for cutoff $= 5$). These two cases are the only times that SSL manages to significantly outperform HYB among all the experiments reported in this article.

The precision values in Table XIV show that HYB always outperforms SSL on the nonrelevant testbed. The improvements range between 5%–25%, and are often statistically significant when five collections are selected.

As in previous experiments with a single retrieval model, HYB and SSL produce comparable results on the representative and gov2 testbeds. On the former testbed (Table XV), HYB has minor advantages, while on the latter (Table XVI), SSL results are slightly better. None of the differences are statistically significant.

7. DISCUSSION

The results produced by our most successful variant of SAFE (HYB) are often comparable with or better than the state-of-the-art SSL across a range of

Table XIV. Precision Values for the Merging Methods on the "Nonrelevant" Testbed (CRCS is used for collection selection. TREC topics 51–100 (short) and their relevance judgments have been used as queries. Collections use different retrieval models (tfidf, KL-Divergence, INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	0.32	0.33	0.35	0.34	0.33	0.33	0.32	0.31
LIN	0.33	0.32	0.31	0.29	0.35	0.34	0.34	0.29
SQRT	0.33	0.32	0.33	0.30	0.34	0.35	0.34	0.31
LOG	0.33	0.33	0.33	0.31	0.34	0.35	0.34	0.31
POW	0.29	0.29	0.14	0.12	0.36	0.33	0.15	0.13
HYB	**0.35**	**0.35**	**0.39**	**0.36**	**0.40***	**0.36**	**0.42**‡	**0.38**†

Table XV. Precision Values for the Merging Methods on the "Representative" Testbed (CRCS is used for collection selection. TREC topics 51–100 (short) and their relevance judgments have been used as queries. Collections use different retrieval models (tfidf, KL-Divergence, INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	0.36	**0.37**	0.37	0.32	0.36	**0.37**	0.36	0.33
LIN	0.36	0.36	0.30	0.28	0.34	0.36	0.27	0.26
SQRT	0.34	0.36	0.30	0.29	0.32	0.34	0.28	0.27
LOG	0.34	0.34	0.30	0.29	0.32	0.34	0.28	0.26
POW	0.33	0.33	0.17	0.16	0.33	0.33	0.18	0.16
HYB	**0.39**	0.36	**0.40**	**0.37**	**0.38**	0.36	**0.37**	**0.35**

scenarios. In single-model environments, HYB significantly outperforms SSL on the trec4 and nonrelevant testbeds. On the remaining testbeds, the results produced by HYB and SSL were never found to be statistically significantly different. Similar observations can be made in multimodel experiments: HYB significantly outperforms SSL on the trec4 and nonrelevant testbeds, and produces comparable results in the other experiments. The only exception is for the relevant testbed, in which SSL manages to produce significantly better results than HYB in two cases. The—relatively—poor performance of HYB on the relevant and gov2 testbeds is consistent with the low R^2 values reported in Table IV. The correlation between R^2 and the merging effectiveness in our experiments suggests that further improvements might be possible by using a more sophisticated regression model, and more accurate collection size estimations.

The precision values in the single-model experiments are usually higher than those in the multimodel scenario. This is not surprising, given that the variant of tfidf assigned to almost one-third of collections in the multimodel experiments is significantly less effective than INQUERY (the retrieval model used

Table XVI. Precision Values for the Merging Methods on the gov2 Testbed (CRCS is used for collection selection. TREC topics 701–750 (short) and their relevance judgments have been used as queries. Collections use different retrieval models (tfidf, KL-Divergence, INQUERY). Parameter η shows the maximum number of answers that each collection may return per query.)

	Three Collections Selected				Five Collections Selected			
	$\eta = 10$		$\eta = 100$		$\eta = 10$		$\eta = 100$	
	P@5	P@10	P@5	P@10	P@5	P@10	P@5	P@10
SSL	**0.14**	**0.14**	0.17	0.14	**0.15**	**0.15**	0.16	0.13
LIN	0.13	0.11	0.11	0.11	0.10	0.09	0.10	0.10
SQRT	0.11	0.11	0.11	0.10	0.10	0.10	0.08	0.08
LOG	0.11	0.10	0.10	0.10	0.10	0.10	0.07	0.08
POW	0.09	0.11	0.06	0.05	0.12	0.12	0.09	0.08
HYB	0.12	0.12	0.17	**0.15**	0.12	0.12	0.16	**0.17**

Table XVII. The Number of Queries (out of 50) for Which SSL Backs Off to CORI in Different Merging Experiments (Parameter η shows the maximum number of answers that each collection may return per query.)

	trec4	Uniform	Relevant	Nonrelevant	Representative	gov2
Three collections selected (single-model)						
$\eta = 10$	44	50	46	48	49	48
$\eta = 100$	01	19	14	15	04	41
Three collections selected (multimodel)						
$\eta = 10$	45	50	47	49	49	48
$\eta = 100$	05	23	15	17	23	39
Five collections selected (single-model)						
$\eta = 10$	44	50	47	49	50	50
$\eta = 100$	01	15	08	11	20	43
Five collections selected (multimodel)						
$\eta = 10$	45	50	47	50	50	50
$\eta = 100$	03	16	10	16	22	42

across all collections in the single-model scenario).[9] A potential direction for future research is to model the search effectiveness of collections using available techniques such as RUM [Si and Callan 2005]. RUM was proposed by Si and Callan [2005] for collection selection. However, with minor modifications it can be also used for merging experiments.

7.1 SSL Backoff Strategy

SSL backs off to the less effective CORI heuristic when there are fewer than three overlapped documents between the results returned by a collection and its sampled documents. The numbers in Tables XVII confirm that such conversion happens frequently with our settings. As expected, the number of overlap documents is higher for single-model experiments, and for longer answer lists ($\eta = 100$). The numbers also reveal that for $\eta = 10$, SSL essentially converts to the less effective CORI.

[9]For example, on a central index of all documents in the trec4 testbed, INQUERY achieves 0.52 for P@5, compared to 0.24 obtained by tfidf.

Table XVIII. Statistical Significance of Differences Between SAFE (HYB) and SSL Across Different Parameter Settings

Parameters	SAFE (Average)	SSL (Average)	Wilcoxon p-value
Single-model experiments			
P@5, CO = 3, $\eta = 10$	0.317	0.304	0.28
P@10, CO = 3, $\eta = 10$	0.286	0.283	0.38
P@5, CO = 5, $\eta = 10$	0.313	0.304	0.50
P@10, CO = 5, $\eta = 10$	0.296	0.283	0.21
P@5, CO = 3, $\eta = 100$	0.314	0.314	0.44
P@10, CO = 3, $\eta = 100$	0.289	0.284	0.30
P@5, CO = 5, $\eta = 100$	0.338	0.313	**0.02**
P@10, CO = 5, $\eta = 100$	0.311	0.277	**0.00**
Multimodel experiments			
P@5, CO = 3, $\eta = 10$	0.299	0.283	0.18
P@10, CO = 3, $\eta = 10$	0.279	0.276	0.66
P@5, CO = 5, $\eta = 10$	0.295	0.286	0.36
P@10, CO = 5, $\eta = 10$	0.278	0.281	0.22
P@5, CO = 3, $\eta = 100$	0.313	0.303	**0.02**
P@10, CO = 3, $\eta = 100$	0.294	0.274	**0.04**
P@5, CO = 5, $\eta = 100$	0.324	0.294	**0.02**
P@10, CO = 5, $\eta = 100$	0.306	0.267	**0.00**
Overall	—	—	**0.00**

7.2 The Impact of Collection Size

SAFE uses the estimated collection size values, to calculate the merging scores. We used capture-history [Shokouhi et al. 2006b], with 140 queries to estimate the size of collections. In our preliminary experiments, we noticed that using better estimation of collection size often leads to higher search effectiveness. However, capture-history usually requires a greater number of sampling queries to produce more accurate estimations. This may not be possible in practice due to efficiency restrictions.

7.3 Overall Significance of Gains

The results in Tables V, VIII, XI, and XIV show significant differences between SAFE (HYB) and SSL. To measure the overall statistical significance of improvements, we first divided the experimental results into 16 sets according to the parameter settings. For each combination, we then merged the results across all the testbeds. Each merged set contains the results of 300 queries (six testbeds, 50 queries each). The results are included in the Table XVIII. It can be seen that, except for two cases, the SAFE results are always better than SSL. The differences are statistically significant according to the Wilcoxon paired test for 6 of 16 combinations. Overall (across 2×16 parameter settings), the Wilcoxon pair test detects significant difference ($p \approx 0.0003$) between SAFE and SSL.

8. CONCLUSIONS AND FUTURE WORK

We have introduced a new result merging algorithm, SAFE, designed for uncooperative environments. The algorithm uses the computed scores on samples

drawn from collections to estimate scores in ranked lists returned by the selected collections. Unlike SSL, our proposed algorithm does not depend on overlap documents.

When collections do not return their document scores, the performance of current FIR merging algorithms may drastically decline. In addition, the state-of-the-art merging methods make assumptions that might not be realistic in a real-life environment. We have shown that SAFE can produce comparable results with SSL, the principal alternative method, across six experimental testbeds. According to our experiments, HYB is the most successful regression model for SAFE.

The advantage of using SAFE is more acute when collections return short answer lists. In such a scenario, the likelihood of visiting a sampled document in collection results is lower. These overlap documents are the main resource used by SSL for computing the merging scores. Therefore, for environments in which collections return short answer lists—and downloading documents on the fly is prohibited—SSL often converts to the less effective CORI.

There are many aspects of SAFE that can be improved in future research. For example, SAFE assumes that all collections use effective models for document retrieval. The results in Section 6.4 show that in the presence of multiple retrieval models, such assumption may lead to significant loss in search effectiveness. Measuring the search effectiveness of FIR collections has been studied in the collection selection literature [Craswell et al. 2000; Nottelmann and Fuhr 2003; Si and Callan 2005]. Such methods can be used with minor modifications for result merging. An alternative solution might be to use a mixture of regression equations to model the search effectiveness of collections. Previous studies [Manmatha et al. 2001] have shown that the scores of relevant and nonrelevant documents, respectively, follow the normal and exponential distributions. Based on such observations, it may be possible to predict the search effectiveness of collections according to their distribution of document scores.

In addition, SAFE tends to overestimate the scores of the top-ranked documents returned from very large collections. For example, the answers returned by a large collection with n low-score sampled documents can be ranked higher than the results of a small collection with n high-score sampled documents. This problem opens an interesting direction for future research. Further, SAFE is only designed for uncooperative environments in which collections do not report their document scores. However, if the document scores are provided by collections, they may help to improve the fitness of SAFE regression models. Our preliminary experiments did not show any benefit for SAFE—in terms of search effectiveness—when the reported document scores were used. However, there is still potential room for improvements, and we aim to modify SAFE to gain more from the published scores.

Finally, in all experiments reported in this article, the size of collection samples was assumed to be fixed and 300 documents. It is interesting to investigate the impact of sample size and other sampling strategies [Azzopardi et al. 2006; Baillie et al. 2006; Caverlee et al. 2006; Shokouhi et al. 2006a] on the effectiveness of final retrieval.

ACKNOWLEDGMENTS

We are grateful to Jamie Callan for his insightful comments on the first draft of this article.

REFERENCES

ABBACI, F., SAVOY, J., AND BEIGBEDER, M. 2002. A methodology for collection selection in heterogeneous contexts. In *Proceedings of the International Conference on Information Technology: Coding and Computing*. IEEE Computer Society Press, Los Alamitos, CA, 529.

ASLAM, J., PAVLU, V., AND SAVELL, R. 2003. A unified model for metasearch, pooling, and system evaluation. In *Proceedings of the 12th International Conference on Information and Knowledge Management* (New Orleans, LA). 484–491.

ASLAM, J. A. AND MONTAGUE, M. 2001. Models for metasearch. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, LA). 276–284.

AVRAHAMI, T., YAU, L., SI, L., AND CALLAN, J. 2006. The FedLemur: Federated search in the real world. *J. Amer. Soc. Inform. Sci. Tech. 57,* 3, 347–358.

AZZOPARDI, L., BAILLIE, M., AND CRESTANI, F. 2006. Adaptive query-based sampling for distributed IR. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, WA). 605–606.

BAILLIE, M., AZZOPARDI, L., AND CRESTANI, F. 2006. Adaptive query-based sampling of distributed collections. In *SPIRE String Processing and Information Retrieval Symposium*. Springer, Glasgow, U.K. 316–328.

BAR-YOSSEF, Z. AND GUREVICH, M. 2006. Random sampling from a search engine's index. In *Proceedings of the 15th International Conference on the World Wide Web* (Edinburgh, U.K.). 367–376.

BERNSTEIN, Y., SHOKOUHI, M., AND ZOBEL, J. 2006. Compact features for detection of near-duplicates in distributed retrieval. In *SPIRE String Processing and Information Retrieval Symposium*. Springer, Glasgow, U.K. 110–121.

BHARAT, K. AND BRODER, A. 1998. A technique for measuring the relative size and overlap of public web search engines. *Comput. Netw. ISDN Syst. 30,* 1-7, 379–388.

CALLAN, J. 2000. Distributed information retrieval. *Advances in Information Retrieval*. Kluwer, Norwell, MA, Chapter 5, 127–150.

CALLAN, J. AND CONNELL, M. 2001. Query-based sampling of text databases. *ACM Trans. Inform. Syst. 19,* 2, 97–130.

CALLAN, J., CONNELL, M., AND DU, A. 1999. Automatic discovery of language models for text databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Philadelphia, PA). 479–490.

CALLAN, J., CROFT, B., AND BROGLIO, J. 1997. TREC and TIPSTER experiments with INQUERY. In *Readings in Information Retrieval*. Morgan Kaufmann, San Francisco, CA, 436–439.

CALLAN, J., LU, Z., AND 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* (Seattle, WA). 21–28.

CAVERLEE, J., LIU, L., AND BAE, J. 2006. Distributed query sampling: A quality-conscious approach. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, WA). 340–347.

CRASWELL, N., BAILEY, P., AND HAWKING, D. 2000. Server selection on the World Wide Web. In *Proceedings of the Fifth ACM Conference on Digital Libraries* (San Antonio) TX. 37–46.

CRASWELL, N., HAWKING, D., AND THISTLEWAITE, P. 1999. Merging results from isolated search engines. In *Proceedings of the 10th Australasian Database Conference*. Springer-Verlag, Auckland, New Zealand, 189–200.

CROFT, B. 2000. Combining approaches to information retrieval. *Advances in Information Retrieval*. Kluwer, Norwell, MA, Chapter 1, 1–36.

DREILINGER, D. AND HOWE, A. 1997. Experiences with selecting search engines using metasearch. *ACM Trans. Inform. Sys. 15,* 3, 195–222.

D'SOUZA, D. AND THOM, J. 1999. Collection selection using n-term indexing. In *Proceedings of the*

Second International Symposium on Cooperative Database Systems for Advanced Applications (CODAS'99). Springer, Wollongong, Australia, 52–63.

D'Souza, D., Thom, J., and Zobel, J. 2004. Collection selection for managed distributed document databases. *Inform. Process. Manage. 40,* 3, 527–546.

Fox, E. and Shaw, J. 1993. Combination of multiple searches. In *Proceedings of the Second Text REtrieval Conference*. NIST Special Publication. National Institute of Science and Technology, Gaithersburg, MD, 243–252.

French, J., Powell, A., Callan, J., Viles, C., Emmitt, T., Prey, K., and Mou, Y. 1999. Comparing the performance of database selection algorithms. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, CA). 238–245.

Fuhr, N. 1999. A decision-theoretic approach to database selection in networked IR. *ACM Trans. Inform. Sys. 17,* 3, 229–249.

Garcia, S., Williams, H., and Cannane, A. 2004. Access-ordered indexes. In *Proceedings of the 27th Australasian Computer Science Conference* (Darlinghurst, Australia). 7–14.

Glover, E., Lawrence, S., Birmingham, W., and Giles, C. 1999. Architecture of a metasearch engine that supports user information needs. In *Proceedings of the 8th ACM CIKM Conference on Information and Knowledge Management* (Kansas City, MO). 210–216.

Gravano, L., Chang, C. K., Garcia-Molina, H., and Paepcke, A. 1997. STARTS: Stanford proposal for Internet meta-searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Tucson, AZ). 207–218.

Gravano, L. and Garcia-Molina, H. 1995. Generalizing GlOSS to vector-space databases and broker hierarchies. In *Proceedings of the 21st International Conference on Very Large Data Bases* (San Francisco, CA). 78–89.

Gravano, L., Garcia-Molina, H., and Tomasic, A. 1994a. The effectiveness of GlOSS for the text database discovery problem. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Minneapolis, MN). 126–137.

Gravano, L., Garcia-Molina, H., and Tomasic, A. 1994b. Precision and recall of GlOSS estimators for database discovery. In *Proceedings of the Third International Conference on Parallel and Distributed Information Systems* (Washington, DC). 103–106.

Gravano, L., Garcia-Molina, H., and Tomasic, A. 1999. GlOSS: Text-source discovery over the Internet. *ACM Trans. Database Sys. 24,* 2, 229–264.

Gravano, L., Ipeirotis, P., and Sahami, M. 2003. Qprober: A system for automatic classification of hidden-Web databases. *ACM Trans. Inform. Sys. 21,* 1, 1–41.

Gross, J. 2003. *Linear Regression*. Springer, Berlin, Germany.

Hedley, Y., Younas, M., James, A., and Sanderson, M. 2004a. A two-phase sampling technique for information extraction from hidden Web databases. In *Proceedings of the 6th Annual ACM International Workshop on Web Information and Data Management* (Washington, DC). 1–8.

Hedley, Y., Younas, M., James, A., and Sanderson, M. 2004b. A two-phase sampling technique to improve the accuracy of text similarities in the categorisation of hidden Web databases. In *Proceedings of the International Conference on Web Informations Systems*. Springer, Brisbane, Australia, 516–527.

Ipeirotis, P. and Gravano, L. 2004. When one sample is not enough: Improving text database selection using shrinkage. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Paris, France). 767–778.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil). 154–161.

Kirsch, T. 2003. Document retrieval over networks wherein ranking and relevance scores are computed at the client for multiple database documents. U.S. Patent 5,659,732.

Lafferty, J. and 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* (New Orleans, LA). 111–119.

Larkey, L., Connell, M., and Callan, J. 2000. Collection selection and results merging with topically organized U.S. patents and TREC data. In *Proceedings of the Ninth International Conference on Information and Knowledge Management* (McLean, VA). 282–289.

LAWRENCE, S. AND GILES, C. 1998. Inquirus, the NECi meta search engine. In *Proceedings of the 7th International Conference on the World Wide Web*. Elsevier Science Publishers B. V., Brisbane, Australia, 95–105.

LEE, J. 1997. Analyses of multiple evidence combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Philadelphia, PA). 267–276.

LILLIS, D., TOOLAN, F., COLLIER, R., AND DUNNION, J. 2006. ProbFuse: A probabilistic approach to data fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, WA). 139–146.

LU, J. AND CALLAN, J. 2002. Pruning long documents for distributed information retrieval. In *Proceedings of the 11th ACM CIKM International Conference on Information and Knowledge Management* (McLean, VA). 332–339.

MANMATHA, R., RATH, T., AND FENG, F. 2001. Modeling score distributions for combining the outputs of search engines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, LA). 267–275.

NG, K. 1998. An investigation of the conditions for effective data fusion in information retrieval. Ph.D. dissertation. Rutgers University, New Brunswick, NJ.

NOTTELMANN, H. AND 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* (Toronto, Ont., Canada). 290–297.

OGILVIE, P. AND CALLAN, J. 2001. The effectiveness of query expansion for distributed information retrieval. In *Proceedings of the 10th ACM CIKM International Conference on Information and Knowledge Management* (Atlanta, GA). 183–190.

OZTEKIN, B., KARYPIS, G., AND KUMAR, V. 2002. Expert agreement and content based reranking in a meta search environment using Mearf. In *Proceedings of the 11th International Conference on the World Wide Web* (Honolulu, HI). 333–344.

PALTOGLOU, G., SALAMPASIS, M., AND SATRATZEMI, M. 2007. Results merging algorithm using multiple regression models. In *Proceedings of the Euorpean Conference on Information Retrieval*. Springer, Rome, Italy, 173–184.

PORTER, M. 1997. An algorithm for suffix stripping. In *Readings in Information Retrieval*. Morgan Kaufmann, San Francisco, CA, 313–316.

POWELL, A. L. AND FRENCH, J. 2003. Comparing the performance of collection selection algorithms. *ACM Trans. Inform. Sys. 21,* 4, 412–456.

RASOLOFO, Y., ABBACI, F., AND SAVOY, J. 2001. Approaches to collection selection and results merging for distributed information retrieval. In *Proceedings of the 10th ACM CIKM International Conference on Information and Knowledge Management* (Atlanta, GA). 191–198.

RASOLOFO, Y., HAWKING, D., AND SAVOY, J. 2003. Result merging strategies for a current news metasearcher. *Inform. Process. Manage. 39,* 4, 581–609.

SELBERG, E. AND ETZIONI, O. 1995. Multi-service search and comparison using the metacrawler. In *Proceedings of the 4th International Conference on the World Wide Web*. Oreilly, Boston, MA.

SELBERG, E. AND ETZIONI, O. 1997. The MetaCrawler architecture for resource aggregation on the web. *IEEE Expert 12,* 1, 8–14.

SHOKOUHI, M. 2007. Central-rank-based collection selection in uncooperative distributed information retrieval. In *Proceedings of the Euorpean Conference on Information Retrieval*. Springer, Rome, Italy, 160–172.

SHOKOUHI, M., SCHOLER, F., AND ZOBEL, J. 2006a. Sample sizes for query probing in uncooperative distributed information retrieval. In *Proceedings of the 8th Asia Pacific Web Conference*. Springer (Harbin, China). 63–75.

SHOKOUHI, M. AND ZOBEL, J. 2007. Federated text retrieval from uncooperative overlapped collections. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands). 495–502.

SHOKOUHI, M., ZOBEL, J., AND BERNSTEIN, Y. 2007. Distributed text retrieval from overlapping collections. In *Proceedings of the 18th Australasian Database Conference*. CRPIT, vol. 63. ACS, Ballarat, Australia, 141–150.

SHOKOUHI, M., ZOBEL, J., SCHOLER, F., AND TAHAGHOGHI, S. 2006b. Capturing collection size for distributed non-cooperative retrieval. In *Proceedings of the 29th Annual International ACM*

SIGIR Conference on Research and Development in Information Retrieval (Seattle, WA). 316–323.

SI, L. AND CALLAN, J. 2002. Using sampled data and regression to merge search engine results. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Tampere, Finland). 19–26.

SI, L. AND CALLAN, J. 2003a. The effect of database size distribution on resource selection algorithms. In *Proeedings of the SIGIR 2003 Workshop on Distributed Information Retrieval* (Toronto, Ont., Canada). 31–42.

SI, L. AND CALLAN, J. 2003b. 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* (Toronto, Ont., Canada). 298–305.

SI, L. AND CALLAN, J. 2003c. A semisupervised learning method to merge search engine results. *ACM Trans. Inform. Sys. 21,* 4, 457–491.

SI, L. AND CALLAN, J. 2004. Unified utility maximization framework for resource selection. In *Proceedings of the 13th ACM CIKM Conference on Information and Knowledge Management* (Washington, DC). 32–41.

SI, L. AND CALLAN, J. 2005. Modeling search engine effectiveness for federated search. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil). 83–90.

SI, L., JIN, R., CALLAN, J., AND OGILVIE, P. 2002. A language modeling framework for resource selection and results merging. In *Proceedings of the 11th ACM CIKM International Conference on Information and Knowledge Management* (McLean, VA). 391–397.

THOMAS, P. AND HAWKING, D. 2007. Evaluating sampling methods for uncooperative collections. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands). 503–510.

VOGT, C. 1999. Adaptive combination of evidence for information retrieval. Ph.D. dissertation. University of California, San Diego, La Jolla, CA.

VOGT, C. AND COTTRELL, G. 1999. Fusion via a linear cmbination of scores. *Inform. Retr. 1,* 3, 151–173.

WANG, Y. AND DEWITT, D. 2004. Computing PageRank in a distributed internet search engine system. In *Proceedings of the 30th International Conference on Very Large Data Bases* (Toronto, Ont., Canada). 420–431.

WU, S. AND MCCLEAN, S. 2006. Performance prediction of data fusion for information retrieval. *Inform. Process. Manage. 42,* 4, 899–915.

WU, S. AND MCCLEAN, S. 2007. Result merging methods in distributed information retrieval with overlapping databases. *Inform. Retr. 10,* 3, 297–319.

XU, J. AND 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* (Melbourne, Australia). 112–120.

XU, J. AND CROFT, B. 1999. Cluster-based language models for distributed retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, CA). 254–261.

XU, J., WU, S., AND LI, X. 2007. Estimating collection size with logistic regression. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands). 789–790.

ZHAI, C. 2001. Notes on the lemur TFIDF model. School of Computer Science. Carnegie Mellon University, Pittsburgh, PA. unpublished report. `www.cs.cmu.edu/ lemur/1.1/tfidf.ps`.

ZOBEL, J. 1997. Collection selection via lexicon inspection. In *Proceedings of the Australian Document Computing Symposium* (Melbourne, Australia). 74–80.

Received October 2006; revised September 2007, May 2008, August 2008; accepted September 2008