The Impact of Database Selection on Distributed Searching

Allison L. Powell James C. French*
Department of Computer Science
University of Virginia
{alp4g|french}@cs.virginia.edu

Jamie Callan†
School of Computer Science
Carnegie Mellon University
callan+@cs.cmu.edu

Margaret Connell‡
Center for Intelligent Information Retrieval
University of Massachusetts
connell@cs.umass.edu

Charles L. Viles§
School of Information and Library Science
University of North Carolina, Chapel Hill
viles@ils.unc.edu

Abstract The proliferation of online information resources increases the importance of effective and efficient distributed searching. Distributed searching is cast in three parts – database selection, query processing, and results merging. In this paper we examine the effect of database selection on retrieval performance. We look at retrieval performance in three different distributed retrieval testbeds and distill some general results. First we find that good database selection can result in better retrieval effectiveness than can be achieved in a centralized database. Second we find that good performance can be achieved when only a few sites are selected and that the performance generally increases as more sites are selected. Finally we find that when database selection is employed, it is not necessary to maintain collection wide information (CWI), e.g. global idf. Local information can be used to achieve superior performance. This means that distributed systems can be engineered with more autonomy and less cooperation. This work suggests that improvements in database selection can lead to broader improvements in retrieval performance, even in centralized (i.e. single database) systems. Given a centralized database and a good selection mechanism, retrieval performance can be improved by decomposing that database conceptually and employing a selection step.

1 Introduction

To date, document retrieval in a distributed environment has been compared to, and performed less effectively than, retrieval in a centralized environment. However, in recent work, Xu and Croft [23] discuss the possibility that retrieval performance in a distributed environment may exceed performance in a centralized environment. In that work, Xu and Croft were pessimistic about the potential to achieve both retrieval efficiency and effectiveness in heterogeneous distributed environments. Instead they focused on document collections created by clustering documents. They achieved good results with this clustering approach; however, clustering requires the cooperation of the sites being searched.

We believe that the potential exists to exceed centralized retrieval performance while maintaining search efficiency even in distributed environments where clustering is not possible or where the composition of document collections is imposed by the owning organization and is outside our control. Document collections may be decomposed according to many different criteria, for example, according to publication source, publication date, or to equalize database size. We will need to engineer retrieval systems that are robust to such decompositions. Accordingly, we need to know how various algorithms behave in such environments. In this paper we report on experiments conducted using three different organizations of documents and examine retrieval performance in each environment.

Our goal for this paper is to investigate the following general questions. For different distributed organizations of documents, how does document retrieval performance compare to centralized performance? What is the overall impact of selecting more or fewer databases to search? What is the impact on document retrieval of disseminating collection information among the databases?

2 Distributed Retrieval, Database Selection and Results Merging

Distributed information retrieval consists of three major steps. First, given a set of databases that may be searched, the database selection step chooses the databases to which queries will be sent. Next, the query is processed at the selected databases, producing a set of individual result-lists. Finally, those result-lists are merged into a single list of documents to be presented to a user.

A number of different approaches for database or collection selection have been proposed and individually evaluated [4, 10, 11, 12, 13, 15, 17, 22, 25]. Three of these approaches, *CORI*[4], *CVV*[25] and *gGlOSS*[11, 12] were evaluated in a common environment by French, *et al.*[3, 7, 8], who found that there was significant room for improvement in all approaches, especially when very few databases were selected.

There has also been attention on results merging or collection fusion. Fox, *et al.*[6] studied the impact of combining the results of multiple query formulations. Belkin, *et al.* [2] examined both combining the results of multiple query formulations and combining retrieval results obtained from multiple retrieval systems. Voorhees, *et al.* [20, 21] proposed a merging approach in which the number of documents retrieved from a database was based on the estimated usefulness of that database. Those documents were then merged using a probabilistic approach. Yager and Rybalov [24] considered the merging prob-

*This work supported in part by DARPA contract N66001-97-C-8542 and NASA GSRP NGT5-50062.

†This work supported in part by NSF grant IIS-9873009.

‡This material is based on work supported in part by the National Science Foundation, Library of Congress and Department of Commerce under cooperative agreement number EEC-9209623. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

§This work supported in part by DARPA contract N66001-97-C-8542.

To be presented at the 23rd ACM-SIGIR International Conference on Research and Development in Information Retrieval, Athens, Greece, July 2000.

lem as stated by Voorhees, *et al.* but enumerated several deterministic merging approaches as an alternative to the original probabilistic approach. Other research efforts have compared multiple merging approaches. Callan, *et al.* [4] considered four different merging approaches in their distributed searching experiments. Craswell, *et al.* [5] proposed two new merging techniques and compared their performance to other published techniques.

In addition to studies of database selection and results merging approaches, there have been broader examinations with the goal of improving distributed retrieval performance. Xu and Callan [22] showed that poor database selection performance hindered distributed retrieval performance, and investigated the use of query expansion and phrases in database selection. Viles and French [9, 19] showed that dissemination of collection information increased retrieval effectiveness. Xu and Croft [23] explored cluster-based language models, investigating different ways to construct database selection indexes.

3 Research Questions

In Section 1, we discussed general questions that prompted this investigation. Here, we state these questions as three hypotheses to clarify the problem.

Hypothesis 1: When very good database selection is employed, distributed retrieval can outperform centralized retrieval in a variety of environments.

Hypothesis 2: It is possible to achieve good document retrieval performance when few databases are selected; however, increasing the number of databases selected will improve performance.

Hypothesis 3: In a distributed environment, the use of collection wide information (CWI) will improve document retrieval performance.

The first hypothesis requires some clarification. We are interested in determining if the use of very good database selection can enable distributed retrieval to outperform centralized retrieval in distributed environments where documents may not have been organized to enhance retrieval. For example, the documents may be organized chronologically or to equalize the size of databases.

4 Experimental Methodology

In the experiments reported here, we examined retrieval performance in both centralized and distributed environments using three different testbeds. We varied the database selection approach, the number of databases searched and the results-merging approach.

We evaluated the impact that these variations had on the final document retrieval results. We searched the highest-ranked databases, merged the returned results, then evaluated the quality of the merged list of documents. Descriptions of the testbeds, details of the selection and merging approaches, and a more detailed description of the evaluation approach are given below.

This work differs from previous research in distributed retrieval in several ways. First, we utilized multiple testbeds with different distributions of relevant documents for our experiments. Second, whereas other efforts have fixed the number of databases selected [3, 22, 23], we study the impact of selecting more or fewer databases. We also consider the combination of both database selection and the dissemination of collection-wide information. Viles and French [19, 9] studied the use of CWI in a distributed environment in which database selection was not used, while Xu and Croft used CWI for all experiments reported in [23].

4.1 Testbeds

We used three different document testbeds in our experiments. All three testbeds are based upon 3 gigabytes of data available to participants in the TREC-4 [14] experiments[1]. The data is spread over several years and comes from seven (7) primary sources: AP Newswire (AP), Wall Street Journal (WSJ), Computer Select (ZIFF), the Patent Office (PAT), San Jose Mercury News (SJMN), Federal Register (FR), and Department of Energy (DOE). The three testbeds were constructed by organizing these documents into databases using the following constraints.

UBC-100 (Uniform-Byte-Count) – Documents from TREC CDs 1, 2, and 3 were organized into document databases of roughly 30 megabytes each, ordered as they appeared on the TREC CDs, and with the restriction that all of the documents in a database were from the same primary source. This testbed contains 100 databases.

SYM-236 (Source-Year-Month) – This testbed was designed to contain a temporal component. Documents were organized into document databases based on the primary source and the month and year of publication. For example, all AP Newswire articles from February of 1988 were placed in the same database. This testbed does not contain documents from DOE or ZIFF documents that appeared on TREC CD 3 (see French *et al.* [8] for details). This testbed contains 236 databases.

UDC-236 (Uniform-Document-Count) – This testbed contains exactly the same documents as *SYM-236*; however, the documents are organized into databases differently. The documents were organized into databases containing roughly 2,900 documents each, ordered as they appeared on the TREC CDs, and with the restriction that all of the documents in a database were from the same primary source. This testbed also contains 236 databases.

SYM-236 and *UDC-236* have been used in evaluations of database selection algorithms [3, 7, 8]. *UBC-100* was used to study the scalability of *CORI* database selection [7].

General characteristics of the testbeds appear in Table 2. This table shows both features of the testbeds and the effects of particular constraints in testbed creation. The *UBC-100* and *UDC-236* testbeds are constructed to contain databases of approximately 30 MB and databases of approximately 2,900 documents, respectively. Depending on individual document size, fixing one of these values can still result in variability in the other. Because there was a temporal component, there was more variability in the sizes of the *SYM-236* databases. For example, there were generally few Patent Office documents in a given month, but there were often many articles from the AP Newswire.

These three testbeds represent three convenient ways to organize documents into databases or to partition a large database into several smaller ones. Xu and Croft [23, p. 256] expressed concern that the distribution of relevant documents in database decompositions such as these may adversely affect the efficiency or effectiveness of distributed retrieval. We discuss this issue in Section 6 and summarize the distribution of relevant documents in the *UBC-100*, *SYM-236*, and *UDC-236* testbeds in Table 6.

4.2 Queries

The TREC data includes a set of fielded *topics*, each of which is a statement of information need. The fields of each topic are different ways of expressing the information need, for example,

[1] Complete maps showing the placement of documents in databases are available. The *UBC-100* map, labelled trec123-100-bysource-callan99.v2a, can be found at http://www.cs.cmu.edu/-callan/Data/. The *SYM-236* and *UDC-236* maps, labelled trec123-236-by_source-by_month and trec123-236-eq_doc_counts respectively, can be found at http://www.cs.virginia.edu/-cyberia/testbed.html.

the Title field contains a very brief description of the information need, while the Concepts field contains words, phrases and proper nouns that might occur in relevant documents. We used the Title field of TREC topics 51-150 to construct a set of short queries, and the Concepts field of the same topics to construct a corresponding set of longer queries. Topics 51-150 were chosen because relevance judgements for those topics were available for all portions of the TREC-4 data (see Table 4 in [8] for an illustration of topic coverage).

4.3 Selection Methodology

We employed two different database selection approaches in our experiments—we chose one achievable approach as a realistic case and one very good, but as yet unachievable approach as a best-case scenario. A comparison of different existing database selection algorithms [3, 7] showed that the *CORI* [4] approach outperformed both *CVV* [25] and *gGlOSS* [11, 12]. As a result, we chose *CORI* as our achievable database selection approach. As our best-case approach, we chose the *RBR* baseline that was used to evaluate the different database selection approaches in French, *et al.* [7].

CORI – Given a set of databases to search, the *CORI* approach creates a *database selection index* in which each database is represented by its terms and their document frequencies *df*. Databases are ranked for a query q by a variant of the Inquery document ranking algorithm. The belief in a database depends upon the query structure, but is usually just the average of the values for each query term [4].

RBR – The *RBR* database rankings were produced using the relevance judgements supplied with the TREC data. Given a TREC query and a set of databases to search, the databases are simply ordered by the number of relevant documents they contain for a query.

In these experiments, we used *RBR* as an oracle for selection; *RBR* provides the best database ordering that is possible given *only* knowledge of where the relevant documents for a query are located. It has no knowledge of document ranking or merging. As a result, there may be situations for which a different ordering of databases produces a better overall document retrieval performance than *RBR*.

4.4 Query Processing

In all scenarios and all experiments, query processing at the databases is performed using Inquery [1]. We used unstructured queries and retrieved 100 documents from each database.

4.5 Merging

We used two different merging approaches in these experiments. The first approach was a simple raw-score merge. When collection-wide information (CWI) is used in a distributed environment, the document scores from different databases are comparable and a raw-score merge is feasible.

Our second merging approach was to use the default Inquery multi-database merging algorithm. This approach uses a combination of the score for the database and the score for the document to estimate a normalized score.

The database score was computed differently for the two database selection approaches. When *CORI* was used for database selection, the normalized database score was computed as follows:

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

where C is the raw database belief score for the database (see any of [3, 7, 8] for the definition of the raw belief score), and C_{max} and C_{min} are the maximum and minimum scores a database could obtain for a particular query. When *RBR* was used for database selection, the normalized database score was computed as:

$$C' = (101 - R)/100 \tag{2}$$

where R is the database rank.

The normalized document score D'' for a document with initial score D was computed as:

$$D' = ((D - D_{min})/(D_{max} - D_{min}) \tag{3}$$

$$D'' = (1.0 \cdot D' + 0.4 \cdot C' \cdot D')/1.4 \tag{4}$$

where D_{max} was the highest score an $ideal$ document could get for that query in that database, and D_{min} was the lowest score a document could get for that query in that database. The normalization of D'' by 1.4 is done to restrict document scores to the range $[0, 1]$.

4.6 The Three Scenarios

To enable a comparison of distributed and centralized performance, and to judge the impact of the use of CWI in distributed retrieval, we used three different scenarios.

`centralized` – For each of the testbeds, all documents are located in a single database.

`dist-CWI` **(collection-wide information)** – For each of the testbeds, documents are distributed into the specified databases. Collection-wide information (CWI) is used for document retrieval. For example, global idf values and all document length values are available. Specifically, for some document d_i and query q_j, the document-query similarity is the same if document d_i is located in a centralized database or in one of the distributed databases. Document scores from different databases are comparable, so a raw-score merge is used.

`dist-LI` **(local information)** – For each of the testbeds, documents are distributed into the specified databases. CWI is *not* available, so only local information from each of the databases is used for document retrieval. Document scores from the different databases are not directly comparable, so the default Inquery merge is used.

4.7 Execution and Evaluation

Given a testbed, a distribution scenario and a selection approach, we used the selection approach to rank all of the databases in the testbed. Then, the top-ranked 2, 5, 10 and 20 databases were considered selected for search. The query used to rank the databases was executed at each selected database and 100 documents were returned from each database. The individual result lists were merged using the merge algorithm specified in the scenario description. The top 100 documents from the merged list were evaluated.

We used the `trec_eval` program to measure precision at document ranks 5, 10, 15, 20, 50 and 100. We used both the paired t-test and paired Wilcoxon test discussed by Hull [16] for significance testing. Due to the presence of ties in our data, we used an alternate formulation of the Wilcoxon test [18]. There was a high degree of agreement between the two tests—the Wilcoxon results are reported here.

5 Results

5.1 Centralized Scenario

Table 1 contains the results for all three testbeds under the `centralized` scenario, where all documents are located in a single database. Note that because they contain exactly the same documents, the results for the *SYM-236* and *UDC-236* testbeds are identical. The documents contained in *SYM-236* and *UDC-236* are a subset of those contained in *UBC-100*—the overall *SYM-236*/*UDC-236* performance results are very close to the *UBC-100* results.

Precision at Rank	*UBC-100*	*SYM-236*	*UDC-236*
5 docs	0.642	0.640	
10 docs	0.609	0.600	
15 docs	0.596	0.593	
20 docs	0.582	0.574	
50 docs	0.538	0.534	
100 docs	0.495	0.484	

Table 1: Average precision values for `centralized`.

5.2 Comparing Distributed and Centralized Performance

Table 3 presents a summary of results for both of the distributed scenarios over all three testbeds. Table 3(a) shows the results for the *UBC-100* testbed, Table 3(b) the *SYM-236* testbed and Table 3(c) the *UDC-236* testbed. Within each sub-table, results for the the two distributed scenarios are shown using both *RBR* and *CORI* selection at 2, 5, 10 and 20 databases selected. Due to space limitations, only the results from the longer (Concept field) queries are reported here. Similar performance trends were seen using the short (Title field) queries; however, the numeric average precision scores were lower.

The typography of Table 3 is used to show the results of a comparison with centralized document retrieval performance. Using a paired Wilcoxon test at p = 0.05, items shown in boldface are significantly better than the corresponding `centralized` performance from Table 1, while italicized items are significantly worse. The default typeface denotes no significant difference. Referring back to Hypothesis 1, we find that when very good (*RBR*) selection is employed, it is possible to exceed centralized performance in all three testbeds. Referring to the first portion of Hypothesis 2, we see that it is possible to meet or exceed centralized performance even when a small number of databases are selected using *RBR* selection. We do, however, see decreased effectiveness at high document ranks when a very small number (2 or 5) of databases are selected. This decreased effectiveness is due to a combination of effects. The first effect is a phenomenon that concerned Xu and Croft [23]—for some queries, there are very few relevant documents to be found in the top-ranked 2 or 5 databases. The second effect is an aspect of the evaluation approach. When very few relevant documents are available in the top 2 or 5 databases, there exist queries for which all available relevant documents may be retrieved in the top 10 or 20 documents. However, because all 100 retrieved documents are evaluated, precision at the 50 or 100 document cutoff for these queries will be very low. These queries can depress the average precision values for high document cutoff values.

The results of the `dist-LI` portions of Table 3 are also shown in Figures 1-6. Figures 1-3 illustrate the potential to exceed `centralized` performance when RBR selection is used.

When currently achievable (*CORI*) selection is employed, the results tend to be significantly worse than the corresponding `centralized` performance (see the right-hand columns of Table 3 and Figures 4-6. However, the results approach `centralized` performance when 20 databases are selected. We discuss the performance when *CORI* selection is employed in more detail in Section 5.4.

5.3 Comparing `dist-CWI` and `dist-LI`

In addition to comparisons of the two distributed approaches to the centralized approach, we were also interested in the relative performance of `dist-CWI` and `dist-LI`. Examining Table 3, we noted that under a strict numeric comparison, `dist-LI` often outperformed `dist-CWI`. The question of whether the difference between `dist-CWI` and `dist-LI` was significant remained. Table 4 repeats the `dist-LI` sections of Table 3 for all three testbeds. Here, however, the numbers in boldface denote cases where the `dist-LI` performance is significantly greater than the corresponding `dist-CWI` performance. In this table, there is no case for which `dist-LI` performance is significantly worse than the corresponding `dist-CWI` performance. For all three testbeds, we find that Hypothesis 3 is false.

This finding appears to contradict the findings of Viles and French [19, 9]; however, there are a number of differences between the experiments reported here and the experiments performed in that work. A discussion of those experimental differences and an analysis of the implications of our results appear in Section 6.

5.4 Number of Databases Searched

In Section 5.2, we noted that it is possible for both `dist-CWI` and `dist-LI` to exceed `centralized` performance, even when a small number of databases were selected for search, confirming the first portion of Hypothesis 2. We now turn our attention to the second portion of that hypothesis. We were interested in the impact of selecting more or fewer databases to search. Examining Table 3, we noted that at each document rank level, increasing the number of databases selected for search tended to (but did not always) improve document retrieval performance. Table 5 addresses the question of whether the observed improvement was significant.

In Table 5, items in boldface type are significantly better (using a paired Wilcoxon test at p = 0.05) than the corresponding item in the column immediately to the left. Bold-italic denotes cases for which an item is significantly better than the corresponding item in *some* column to the left. The default typeface denotes no significant difference. Note that this typography convention is different from the convention used in Tables 3 and 4.

There are a number of interesting things to observe in Table 5. First, when *CORI* is used for selection, selecting a larger number of databases for search tends to be advantageous. This is understandable given that, while its database selection performance is good, *CORI* is not guaranteed to select databases with the most (or even any) relevant documents. In this case, selecting more databases increases the chances of selecting a relevant-rich database. The beneficial effect of selecting additional databases when *CORI* selection is employed is illustrated in Figures 4-6.

When *RBR* is used for selection, the greatest improvement can be seen when 5 or 10 databases are selected (instead of 2). This can be seen in both Table 5 and Figures 1-3. This may be due to the effect discussed in Section 5.2—there are queries for which there are few relevant documents to be found in the top 2 databases. Searching a larger number of databases increases the number of available relevant documents. The lesser improvement when 20 databases are selected can be explained by a similar phenomenon—there also exist queries for which many relevant documents can be found in the top 10 selected databases. For these queries, searching a larger number of databases does not provide a large benefit.

More isn't always better

Finally, we should point out that while searching additional databases tended to improve retrieval performance in Table 5, there are limits to that trend. When all databases that contain relevant documents have been selected, no additional improvement will be seen.

In fact, beyond a certain point, searching additional databases may degrade performance. For example, consider the `dist-CWI` portions of Table 3 when *RBR* is used for database selection. For all three testbeds, there are numerous cases for which `dist-CWI` outperforms `centralized`. However, given the construction of `dist-CWI`, when all databases are

Testbed	Total Docs.	Total DB	Number of Docs. per DB			Number of Bytes. per DB		
			Minimum	Average	Maximum	Minimum	Average	Maximum
UBC-100	1,078,166	100	752	10,782	39,723	28,070,646	33,365,514	41,796,822
SYM-236	691,058	236	1	2,928	8,302	7,668	11,789,423	34,782,134
UDC-236	691,058	236	2,891	2,928	3,356	7,138,629	11,789,423	133,206,035

Table 2: Summary statistics for the testbeds.

UBC 100-collection testbed

	Precision at Rank	RBR selection				CORI selection			
		2 sel	5 sel	10 sel	20 sel	2 sel	5 sel	10 sel	20 sel
dist-CWI	5 docs	0.670	0.652	0.670	0.670	*0.509*	*0.528*	*0.570*	*0.602*
	10 docs	0.633	**0.651**	**0.647**	**0.661**	*0.477*	*0.512*	*0.551*	0.588
	15 docs	0.608	**0.637**	**0.633**	**0.643**	*0.451*	*0.498*	*0.531*	*0.567*
	20 docs	0.586	**0.623**	**0.623**	**0.628**	*0.430*	*0.481*	*0.520*	*0.553*
	50 docs	*0.481*	0.551	**0.571**	**0.582**	*0.340*	*0.416*	*0.462*	*0.508*
	100 docs	*0.375*	*0.468*	0.508	**0.523**	*0.259*	*0.348*	*0.398*	*0.452*
dist-LI	5 docs	**0.686**	**0.694**	0.680	0.682	*0.540*	*0.571*	*0.586*	0.610
	10 docs	**0.656**	**0.683**	**0.685**	**0.691**	*0.501*	0.561	*0.567*	0.595
	15 docs	**0.629**	**0.665**	**0.669**	**0.682**	*0.475*	*0.535*	*0.558*	0.593
	20 docs	0.605	**0.653**	**0.659**	**0.673**	*0.454*	*0.508*	*0.546*	0.579
	50 docs	*0.490*	**0.570**	**0.606**	**0.616**	*0.361*	*0.432*	*0.485*	0.529
	100 docs	*0.378*	0.481	**0.531**	**0.556**	*0.265*	*0.358*	*0.412*	*0.473*

(a)

SYM 236-collection testbed

	Precision at Rank	RBR selection				CORI selection			
		2 sel	5 sel	10 sel	20 sel	2 sel	5 sel	10 sel	20 sel
dist-CWI	5 docs	0.646	0.680	0.674	**0.690**	*0.483*	*0.546*	*0.554*	*0.592*
	10 docs	0.613	**0.643**	**0.653**	**0.673**	*0.464*	*0.499*	*0.527*	*0.555*
	15 docs	0.569	0.620	**0.635**	**0.653**	*0.423*	*0.469*	*0.510*	*0.540*
	20 docs	0.545	0.598	**0.621**	**0.636**	*0.397*	*0.448*	*0.493*	*0.535*
	50 docs	*0.431*	0.513	0.542	**0.580**	*0.287*	*0.368*	*0.416*	*0.472*
	100 docs	*0.309*	*0.414*	0.469	**0.506**	*0.195*	*0.287*	*0.343*	*0.404*
dist-LI	5 docs	**0.700**	**0.718**	**0.704**	**0.726**	*0.538*	*0.568*	*0.554*	0.624
	10 docs	**0.647**	**0.682**	**0.696**	**0.705**	*0.480*	*0.537*	*0.553*	0.586
	15 docs	0.608	**0.656**	**0.663**	**0.698**	*0.435*	*0.506*	*0.536*	0.570
	20 docs	0.572	**0.626**	**0.652**	**0.677**	*0.403*	*0.483*	*0.520*	0.556
	50 docs	*0.440*	0.532	**0.569**	**0.603**	*0.296*	*0.379*	*0.435*	*0.495*
	100 docs	*0.316*	*0.427*	0.484	**0.528**	*0.200*	*0.297*	*0.356*	*0.420*

(b)

UDC 236-collection testbed

	Precision at Rank	RBR selection				CORI selection			
		2 sel	5 sel	10 sel	20 sel	2 sel	5 sel	10 sel	20 sel
dist-CWI	5 docs	**0.726**	**0.700**	**0.708**	**0.708**	*0.480*	*0.506*	*0.546*	*0.557*
	10 docs	**0.658**	**0.684**	**0.680**	**0.693**	*0.427*	*0.479*	*0.499*	*0.531*
	15 docs	0.604	**0.653**	**0.669**	**0.679**	*0.384*	*0.444*	*0.491*	*0.514*
	20 docs	0.574	**0.623**	**0.650**	**0.666**	*0.346*	*0.419*	*0.468*	*0.493*
	50 docs	*0.386*	0.512	**0.565**	**0.598**	*0.228*	*0.315*	*0.384*	*0.430*
	100 docs	*0.246*	*0.380*	0.462	**0.515**	*0.143*	*0.221*	*0.293*	*0.359*
dist-LI	5 docs	**0.718**	**0.722**	**0.732**	**0.732**	*0.501*	*0.508*	*0.549*	*0.561*
	10 docs	**0.662**	**0.692**	**0.707**	**0.699**	*0.443*	*0.499*	*0.541*	0.547
	15 docs	0.620	**0.676**	**0.681**	**0.693**	*0.400*	*0.460*	*0.513*	*0.530*
	20 docs	0.574	**0.638**	**0.665**	**0.668**	*0.362*	*0.431*	*0.493*	*0.518*
	50 docs	*0.396*	0.520	**0.575**	**0.614**	*0.236*	*0.325*	*0.391*	*0.455*
	100 docs	*0.249*	*0.388*	0.471	**0.526**	*0.147*	*0.228*	*0.303*	*0.370*

(c)

Table 3: Average precision achieved in `dist-CWI` and `dist-LI` for *UBC-100*, *SYM-236* and *UDC-236* testbeds. Typeface changes reflect a comparison with `centralized` performance (bold = significantly better, italics = significantly worse).

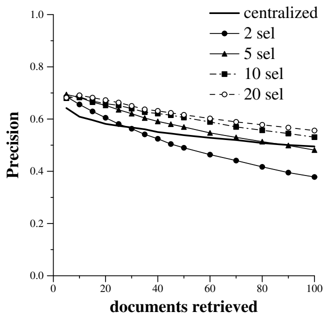

Figure 1: UBC-100 testbed, RBR sel.

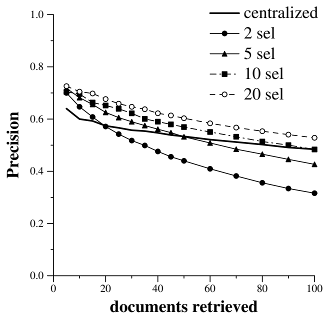

Figure 2: SYM-236 testbed, RBR sel.

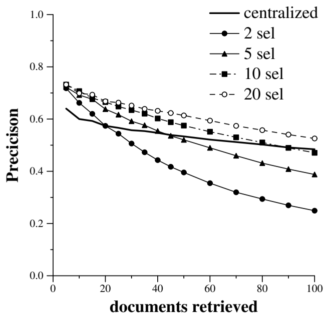

Figure 3: UDC-236 testbed, RBR sel.

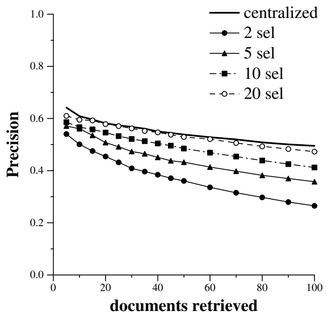

Figure 4: UBC-100 testbed, CORI sel.

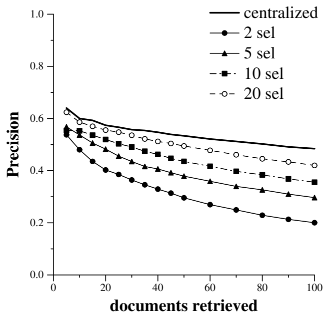

Figure 5: SYM-236 testbed, CORI sel.

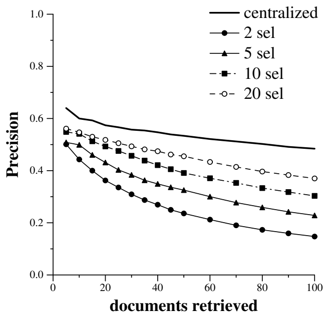

Figure 6: UDC-236 testbed, CORI sel.

selected the performance (when up to 100 documents are retrieved) will be exactly that of centralized. At some point between selecting 20 databases and selecting all of them, performance began to degrade.

6 Discussion

6.1 CWI and Merging Analysis

An issue that deserves immediate attention is the apparent contradiction of this work and the work of Viles and French[9, 19]. Based on the work of Viles and French, we expected that Hypothesis 3 would be true (i.e., the use of CWI would improve distributed retrieval performance); however, the `dist-LI` results were significantly better than the `dist-CWI` results.

Our initial reaction was that the difference in the `dist-CWI` and `dist-LI` results was due to the difference in the merging step for the two scenarios. `dist-CWI` used a raw-score merge while `dist-LI` used the default *CORI* merge. We speculated that the incorporation of the database score into the *CORI* merge contributed to the performance difference. Therefore, we replaced the raw score merge used in the `dist-CWI` case with the *CORI* merge, creating `dist-CWI-CM`.

When we compared `dist-CWI-CM` and `dist-CWI`, we found that the performance of `dist-CWI` was either the same as or better than the performance of `dist-CWI-CM`, eliminating the merge explanation. However, there are additional differences between the `dist-CWI` and `dist-LI` experiments and between our experiments and those of Viles and French that help explain the results. First, Viles and French were investigating a different problem. They showed that when a query was broadcast to *all* databases, a raw score merge using CWI is better than raw score merge using only local database information. Second, and related to the first point, `dist-CWI` and `dist-LI` represent different ways to make the document scores from different databases comparable. In `dist-CWI`, the use of CWI makes the document scores directly comparable. The D'' normalization step in `dist-LI` also makes the document scores from different databases comparable. In `dist-LI`, the general interpretation is that documents that scored well within their database and that also came from highly-ranked databases should be ranked highly.

However, these differences should not be allowed to distract from the take-home message. Given a `dist-CWI` or `dist-LI` scenario, very good database performance enables very good document retrieval performance. Currently-achievable database selection performance enables document retrieval performance on par with centralized; better selection can enable distributed performance to exceed centralized.

6.2 Distribution of Relevant Documents

Table 6 summarizes the distributions of relevant documents in the *UBC-100*, *SYM-236* and *UDC-236* testbeds. The number of databases containing relevant documents, and the distribution of those relevant documents is query-dependent. The values shown here are average values for queries 51-150. The first data column, labelled $Average\ n^*$ is simply the average (over all 100 queries) of the number of databases that contain at least one relevant document. The remaining three data columns summarize the distribution of relevant documents. For each query, we divided the total number of relevant documents by the n^* value for that query. We report the minimum, maximum and average values for that ratio.

Note that in Table 6, the *UBC-100* testbed tends to have both more databases with relevant documents and more relevant documents per database. However, due to the constraints employed when creating the *UBC-100* testbed, there are also more documents per database. Also note that relevant documents are more evenly distributed in the *UDC-236* testbed than in the other two.

Xu and Croft [23] expressed concern that the distribution of relevant documents in distributed databases organized by publication source or database size might hinder distributed retrieval performance. For our three testbeds, the distribution of relevant documents does not appear to have had a large impact on the overall retrieval performance. Each testbed has different relevant document distribution characteristics; however, the overall performance for the three testbeds was similar (see Table 3).

	Precision at Rank	RBR selection 2 sel	RBR selection 5 sel	RBR selection 10 sel	RBR selection 20 sel	CORI selection 2 sel	CORI selection 5 sel	CORI selection 10 sel	CORI selection 20 sel
UBC-100	5 docs	0.686	**0.694**	0.680	0.682	**0.540**	**0.571**	0.586	0.610
	10 docs	0.656	**0.683**	**0.685**	**0.691**	**0.501**	**0.561**	0.567	0.595
	15 docs	**0.629**	**0.665**	**0.669**	**0.682**	**0.475**	**0.535**	**0.558**	**0.593**
	20 docs	**0.605**	**0.653**	**0.659**	**0.673**	**0.454**	**0.508**	**0.546**	**0.579**
	50 docs	0.490	**0.570**	**0.606**	**0.616**	**0.361**	**0.432**	**0.485**	**0.529**
	100 docs	0.378	**0.481**	**0.531**	**0.556**	**0.265**	**0.358**	**0.412**	**0.473**
SYM-236	5 docs	**0.700**	**0.718**	**0.704**	**0.726**	**0.538**	0.568	0.554	**0.624**
	10 docs	**0.647**	**0.682**	**0.696**	**0.705**	0.480	**0.537**	**0.553**	**0.586**
	15 docs	**0.608**	**0.656**	**0.663**	**0.698**	0.435	**0.506**	**0.536**	**0.570**
	20 docs	**0.572**	**0.626**	**0.652**	**0.677**	0.403	**0.483**	**0.520**	0.556
	50 docs	0.440	**0.532**	**0.569**	**0.603**	**0.296**	0.379	**0.435**	0.495
	100 docs	**0.316**	**0.427**	**0.484**	**0.528**	**0.200**	**0.297**	**0.356**	**0.420**
UDC-236	5 docs	0.718	0.722	0.732	0.732	0.501	0.508	0.549	0.561
	10 docs	0.662	0.692	**0.707**	0.699	0.443	0.499	**0.541**	0.547
	15 docs	0.620	**0.676**	0.681	0.693	**0.400**	0.460	**0.513**	0.530
	20 docs	0.574	**0.638**	0.665	0.668	**0.362**	0.431	**0.493**	**0.518**
	50 docs	**0.396**	0.520	0.575	**0.614**	**0.236**	0.325	0.391	**0.455**
	100 docs	0.249	**0.388**	**0.471**	**0.526**	0.147	**0.228**	**0.303**	**0.370**

Table 4: Is `dist-LI` significantly better than `dist-CWI`? Typeface changes — bold indicates `dist-LI` significantly better).

	Precision at Rank	RBR selection 2 sel	RBR selection 5 sel	RBR selection 10 sel	RBR selection 20 sel	CORI selection 2 sel	CORI selection 5 sel	CORI selection 10 sel	CORI selection 20 sel
UBC-100	5 docs	0.686	0.694	0.680	0.682	0.540	0.571	***0.586***	***0.610***
	10 docs	0.656	0.683	0.685	0.691	0.501	**0.561**	***0.567***	***0.595***
	15 docs	0.629	**0.665**	***0.669***	***0.682***	0.475	**0.535**	***0.558***	**0.593**
	20 docs	0.605	**0.653**	***0.659***	***0.673***	0.454	**0.508**	***0.546***	**0.579**
	50 docs	0.490	**0.570**	**0.606**	***0.616***	0.361	**0.432**	**0.485**	**0.529**
	100 docs	0.378	**0.481**	**0.531**	**0.556**	0.265	**0.358**	**0.412**	**0.473**
SYM-236	5 docs	0.700	0.718	0.704	0.726	0.538	0.568	0.554	***0.624***
	10 docs	0.647	**0.682**	***0.696***	***0.705***	0.480	**0.537**	***0.553***	***0.586***
	15 docs	0.608	**0.656**	***0.663***	**0.698**	0.435	**0.506**	**0.536**	**0.570**
	20 docs	0.572	**0.626**	**0.652**	**0.677**	0.403	**0.483**	**0.520**	**0.556**
	50 docs	0.440	**0.532**	**0.569**	**0.603**	0.296	**0.379**	**0.435**	**0.495**
	100 docs	0.316	**0.427**	**0.484**	**0.528**	0.200	**0.297**	**0.356**	**0.420**
UDC-236	5 docs	0.718	0.722	0.732	0.732	0.501	0.508	***0.549***	***0.561***
	10 docs	0.662	**0.692**	***0.707***	***0.699***	0.443	**0.499**	**0.541**	***0.547***
	15 docs	0.620	**0.676**	***0.681***	***0.693***	0.400	**0.460**	**0.513**	***0.530***
	20 docs	0.574	**0.638**	**0.665**	***0.668***	0.362	**0.431**	**0.493**	**0.518**
	50 docs	0.396	**0.520**	**0.575**	**0.614**	0.236	**0.325**	**0.391**	**0.455**
	100 docs	0.249	**0.388**	**0.471**	**0.526**	0.147	**0.228**	**0.303**	**0.370**

Table 5: The impact of selecting more or fewer collections for search using scenario `dist-LI`. Bold = significantly better than item directly to left, Bold italic = better than *some* item to left.

Testbed	Average n*	Rel. docs per DB Min.	Avg.	Max.
UBC-100	51.6	1.9	8.7	26.8
SYM-236	76.7	1.3	5.1	15.4
UDC-236	111.0	1.1	3.4	8.8

Table 6: Summary statistics for the testbeds.

6.3 An Alternate Interpretation of `dist-CWI`

In these experiments, the `dist-CWI` scenario was considered in the context of an existing distributed environment. However, given the potential of distributed retrieval to outperform centralized retrieval, we consider an alternate interpretation.

Given a centralized database, the documents in that database could be conceptually organized into a "distributed" arrangement. The physical storage and organization of the documents need not change, but each document would be assigned to a conceptual "pseudo-database". Queries could be handled at the database as usual, producing a single result list. The impact of the conceptual "distributed" organization could be realized as a post-processing step. Given a query, a database selection step could be added, performed on the pseudo-databases. Documents from the selected pseudo-databases would be declared eligible for retrieval. Only eligible documents would be presented to a user. Our results from the `dist-CWI` experiments suggest that this post-processing step could improve the quality of the result-list.

In order to achieve this, however, improvements in database selection performance are necessary. With existing approaches, we have seen that it is possible to equal centralized performance. It was only with the best-case selection scenario that we saw improvements over centralized performance.

7 Conclusions

In this paper we have reported experiments that support the following conclusions.

- When very good selection is employed, distributed retrieval can outperform centralized.

- It is possible to achieve good document retrieval performance by selecting a small number of heterogeneous databases. Selecting more databases does improve performance (up to a point).
- Given very good selection, conceptually decomposing a centralized database and interposing a selection step has the potential to improve performance.
- The use of collection-wide information is a complex issue. Given the scenario in the bullet above, the straightforward approach of using already-available CWI plus a raw score merge produces good results. However, given a distributed environment, using local information works well if selection is employed and the document scores are suitably normalized before merging is performed.

By examining three different document testbeds we have also shown that these conclusions have wide applicability. There are several implications to these conclusions. First, centralized performance is not necessarily the gold standard that we should be aiming for. It is possible for distributed searches to achieve better retrieval performance.

Second, we can get good retrieval performance when only a few database are selected. This implies that distributed searching with good database selection should scale well.

Third, we can conceptually decompose a single database into subcollections and by introducing a selection step it is possible to achieve better performance than by searching (ranking) the entire database. So we find that selection plus ranking has the potential to improve the effectiveness of ranking alone. Moreover, we can use a simple raw score merge in this case and nothing more elaborate.

Fourth, local information is adequate for good retrieval performance when good database selection is employed. This means that it is unnecessary to disseminate collection-wide information when selection is a part of the search strategy.

We set out to examine the effect of database selection on end-user retrieval performance. Previous work focussed on explicit evaluation of the database selection technique. Our work sought to determine the degree to which database selection would have an impact on retrieval performance. We believe that we met our goal and have provided concrete conclusions that can usefully guide the engineering of large-scale distributed information retrieval systems.

References

[1] J. Allan, J. P. Callan, W. B. Croft, L. Ballesteros, D. Byrd, R. Swan, and J. Xu. INQUERY does battle with TREC-6. In *The Sixth Text REtrieval Conference (TREC-6)*.

[2] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the Evidence of Multiple Query Representations for Information Retrieval. *Information Processing and Management*, 31(4):431–448, 1995.

[3] J. Callan, A. L. Powell, J. C. French, and M. Connell. The Effects of Query-Based Sampling on Automatic Database Selection Algorithms. Technical Report CMU-LTI-00-162, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2000.

[4] J. P. Callan, Z. Lu, and W. B. Croft. Searching Distributed Collections with Inference Networks. In *Proc. SIGIR'95*, pages 21–29, 1995.

[5] N. Craswell, D. Hawking, and P. Thistlewaite. Merging Results from Isolated Search Engines. In *Proc. of the Tenth Australasian Database Conf.*, pages 189–200, 1999.

[6] E. A. Fox, M. P. Koushik, J. Shaw, R. Modlin, and D. Rao. Combining Evidence from Multiple Searches. In *The First Text Retrieval Conference (TREC-1)*, pages 319–328, November 1992.

[7] J. C. French, A. L. Powell, J. Callan, C. L. Viles, T. Emmitt, K. J. Prey, and Y. Mou. Comparing the Performance of Database Selection Algorithms. In *Proc. SIGIR'99*, pages 238–245, 1999.

[8] J. C. French, A. L. Powell, C. L. Viles, T. Emmitt, and K. J. Prey. Evaluating Database Selection Techniques: A Testbed and Experiment. In *Proc. SIGIR'98*, pages 121–129, 1998.

[9] J. C. French and C. L. Viles. Ensuring Retrieval Effectiveness in Distributed Digital Libraries. *Journal of Visual Communication and Image Representation*, 7(1):61–73, 1996.

[10] N. Fuhr. A Decision-Theoretic Approach to Database Selection in Networked IR. *ACM Transactions on Information Systems*, 17(3):229–249, July 1999.

[11] L. Gravano and H. Garcia-Molina. Generalizing GlOSS to Vector-Space Databases and Broker Hierarchies. In *Proc. VLDB'95*, 1995.

[12] L. Gravano, H. Garcia-Molina, and A. Tomasic. GlOSS: Text-source discovery over the internet. *ACM Transactions on Database Systems*, 24(2):229–264, June 1999.

[13] L. Gravano, H. Garcia-Molina, and A. Tomasic. The Effectiveness of GlOSS for the Text Database Discovery Problem. In *SIGMOD94*, pages 126–137, May 1994.

[14] D. Harman. Overview of the Fourth Text Retrieval Conference (TREC-4). In *Proceedings of the Fourth Text Retrieval Conference (TREC-4)*, 1996.

[15] D. Hawking and P. Thistlewaite. Methods for Information Server Selection. *ACM Transactions on Information Systems*, 17(1):40–76, January 1999.

[16] D. Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. In *Proc. SIGIR'93*, pages 329–338, 1993.

[17] A. Moffat and J. Zobel. Information Retrieval Systems for Large Document Collections. In *Proceedings of the Third Text Retrieval Conference (TREC-3)*, pages 85–94, 1995.

[18] R. L. Ott. *An Introduction to Statistical Methods and Data Analysis*. Duxbury Press, 4th. edition, 1993.

[19] C. L. Viles and J. C. French. Dissemination of Collection Wide Information in a Distributed Information Retrieval System. In *Proc. SIGIR'95*, pages 12–20, July 1995.

[20] E. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning Collection Fusion Strategies. In *Proc. SIGIR'95*, pages 172–179, 1995.

[21] E. Voorhees, N. K. Gupta, and B. Johnson-Laird. The Collection Fusion Problem. In *Proceedings of the Third Text Retrieval Conference (TREC-3)*, pages 95–104, 1995.

[22] J. Xu and J. Callan. Effective Retrieval with Distributed Collections. In *Proc. SIGIR'98*, pages 112–120, 1998.

[23] J. Xu and W. B. Croft. Cluster-based Language Models for Distributed Retrieval. In *Proc. SIGIR'99*, pages 254–261, 1999.

[24] R. R. Yager and A. Rybalov. On the fusion of documents from multiple collection information retrieval systems. *Journal of the American Society of Information Science*, 49(13):1177–1184, 1998.

[25] B. Yuwono and D. L. Lee. Server Ranking for Distributed Text Retrieval Systems on Internet. In *Proc. Fifth Intl. Conf. on Database Systems for Advanced Applications*, pages 41–49, April 1997.