Effective Retrieval with Distributed Collections

Jinxi Xu and Jamie Callan
Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts, Amherst
Amherst, MA 01003-4610, USA
www.cs.umass.edu/~xu   www.cs.umass.edu/~callan

Abstract This paper evaluates the retrieval effectiveness of distributed information retrieval systems in realistic environments. We find that when a large number of collections are available, the retrieval effectiveness is significantly worse than that of centralized systems, mainly because typical queries are not adequate for the purpose of choosing the right collections. We propose two techniques to address the problem. One is to use phrase information in the collection selection index and the other is query expansion. Both techniques enhance the discriminatory power of typical queries for choosing the right collections and hence significantly improve retrieval results. Query expansion, in particular, brings the effectiveness of searching a large set of distributed collections close to that of searching a centralized collection.

1 Introduction

In today's network environments, information is highly distributed. The Internet or World Wide Web, for example, contains thousands of collections. It is impractical to create a single centralized index that includes all the documents in all the collections. It would be slow to search such a gigantic index and wasteful of computer resources due to replication of information. Even if it is computationally feasible to do so, it would still be problematic because many collections (sites) are protected by copyrights. IR research, which traditionally studied centralized collections, faces new challenges in the distributed environment. As a result, distributed information retrieval has become an important area in IR research in recent years.

The most important issue in searching a set of distributed collections is how to find the right collections to search for a query. The sheer number of collections in a realistic environment makes exhaustive processing of every collection infeasible. Furthermore, many collections are proprietary and may charge users for searching them. The only method for timely and economic retrieval is to constrain the scope of searching to those collections which are likely to contain relevant documents for a query. For this purpose, the distributed IR system used in this paper adopted a widely used technique: It creates a *collection selection index*. The collection selection index consists of a set of *virtual documents*, each of which is a light-weight representation of a collection. Specifically, the virtual document for a collection is simply a complete list of words in that collection and their frequencies (numbers of containing documents). When a query is posted, the system first compares it with the virtual documents to decide which collections are most likely to contain relevant documents for the query. Document retrieval will only take place at such collections. A complete discussion of the system can be found in [4]. The virtual documents are a very concise representation, requiring less than 1.0% of space taken by the underlying document collections [4]. Furthermore, the process of creating the virtual documents is completely automatic and hence scales well in a realistic environment.

The primary concern of this paper is the retrieval effectiveness of distributed information retrieval. Two things make this work different from previous work in this area. Firstly, the sets of distributed collections used in this paper are far more realistic than those used in previous studies. Our sets of collections are large, each consisting of up to 107 collections. In contrast, those used in previous studies usually consisted of a very small number of collections [4, 11, 20]. As we will see later in this paper, the number of collections in a distributed environment can significantly affect the retrieval effectiveness. Secondly, our evaluation is based on actual relevance judgment data (from the TREC conferences [12]). This is in contrast to some studies in which evaluation was performed in absence of actual relevance data. Gravano, for example, used Boolean satisfaction of a query by a document as "relevance" [11].

We find that for typical queries, the effectiveness of searching a large set of distributed collections is significantly (about 30%) worse than that of searching a single centralized collection. The primary cause is that typical queries, though adequate for document retrieval, are not very suitable for collection selection. Fortunately, this problem can be largely solved through query expansion. When typical queries are expanded by local context analysis [22], the retrieval effectiveness of searching a large set of distributed collections can rival that of searching a centralized collection even when a small number of collections are searched. We also considered using phrases as well as single words in the collection selection index, in the hope of partially compensating for the loss of document boundary information in the collection selection index. Using phrases also improves retrieval effectiveness.

The rest of the paper is structured as following: Section 2 discusses related work. Section 3 discusses our motivation and techniques in more detail. Section 4 describes the sets of collections used for evaluation and how experiments were carried out. Sections 5 to 9 present experimental results and provide detailed analysis. Section 10 draws conclusions and points out future work.

2 Related Work

There have been a number of studies concerning retrieval effectiveness in a distributed environment. Callan, Lu and Croft compared the retrieval effectiveness of searching a set of distributed collections with that of searching a centralized collection [4]. They found no significant difference in retrieval performance between distributed and centralized searching when about half of the collections were searched for a query on average. Since the total number of collections was small (17) and the percentage of collections searched was high, their results may not reflect the true retrieval performance in a realistic distributed environment.

Gravano, et al, evaluated the capability of the GlOSS system for choosing the right collections for a query [11]. Their experiments were carried out on a set of 6 collections [11]. Without human judgments of relevance, their evaluation was based on how many documents in a collection satisfy a query in the Boolean sense. If many documents in a collection contain the words "white" and "house", it is considered a hit for an information need about the presidential mansion of the United States expressed as a Boolean query "white AND house", even if none of documents is relevant. It is hard to gauge the true value of their system because of this flaw in evaluation. The work was later extended to the Vector Space Model, but the lack of relevance judgments was still a problem [10].

Voorhees, et al, evaluated the effectiveness of searching a set of distributed collections from the perspective of collection fusions [20]. They found that when the number of documents to be retrieved from each collection for a query is properly tuned based on relevance judgments for a set of training queries, the effectiveness of searching a distributed set of collections is less than 10% worse than that of searching a centralized collection. A problem with their technique is that it requires relevance judgment for a set of training queries, which can be very expensive to obtain in practice. The other problem is that they only used 5 collections.

Viles and French studied the impact of inverse document frequency (idf) estimation on the performance of distributed retrieval and proposed a method to disseminate collection wide information periodically to achieve better estimation of idf statistics in a dynamic distributed environment [19].

Moffat, et al, used a centralized index on blocks of documents [16]. In that study, an initial search on the centralized index returns the highly ranked blocks for a query. A second retrieval returns the highly ranked documents from the highly ranked blocks. The retrieval performance is highly dependent on the block size. Severe degradation in performance was observed when each block contains 1000 documents.

Despite the difference in terminology, the general concept of a collection selection index has occurred in many studies [15, 11, 10, 4, 6, 1]. A collection selection index typically contains a list of textual objects, each of which is a concise description about the content of a collection. In some studies the collection selection index assumes a more complex hierarchical structure [10, 1]. A common technique to describe the content of a collection is to use the words that occur in the collection and their frequency of occurrences. This technique was used by Gravano, et al, in the GlOSS project [11, 10] and later proposed as part of the STARTS standard [9]. The major advantage of this technique is that it is cheap to obtain and scales well in a realistic distributed environment. This technique was also used by Callan, et al, in [4]. The primary difference from GlOSS was that collection selection in [4] is based on a probabilistic model, the Inference Network model [18]. Other techniques for describing the content of a collection are more expensive. In WAIS, a piece of natural language text had to be manually written to describe the content of a collection [15]. In NertSerf, the descriptions of the collections were also created manually and represented as frames [6].

Chamis used switching vocabularies for selection of online text databases [7]. In that study, manual thesauri were used to reconcile the vocabulary difference among the databases.

Fuhr proposed a decision-theoretic approach to collection selection [8]. While most studies only considered how likely a collection contains relevant documents to a query or how many documents in the collection may be relevant, Fuhr's approach also considered the costs (money and computer resources) associated with searching the collection.

3 Motivation

Collection selection is based on how well a virtual document satisfies the information need expressed by a query. A virtual document is a list of words and their frequencies in the corresponding collection. More formally, the virtual document for a collection C is

$$VD(C) = \{< w_i, f_i >\}$$

where w_i's are words occurring in C, and f_i is the document frequency of w_i in C. Suppose we are searching a set of collections C_1, C_2, ..., C_n. Given a query Q, the virtual documents $VD(C_i)$'s are treated as normal documents and are ranked for Q based on a probabilistic model. The top ranked m collections are chosen for retrieval. The algorithm for ranking the collections is fully described in [4, 18]. To understand the arguments in this paper, it is sufficient to know that the process is analogous to that of ranking the documents in an ordinary document collection.

We conjecture that the above approach will do a poor job of ranking the collections for typical queries. A typical query is a short casual description of an information need typically used for ad hoc retrieval in an interactive environment. Examples are "the White House", "high blood pressure", and "forest destruction and weather". These are opposed to long elaborate queries typical in routing environments. The reason for the poor ranking of the collections is not that the ranking algorithm is bad. In fact, the quality of the algorithm for document ranking is well documented [18, 5]. The reason is that typical queries do not provide enough information for collection ranking under the chosen representation of the collections (as virtual documents), no matter how good the ranking algorithm is. Take the query "the White House" for an example. The occurrences of the words "white" and "house" almost uniformly spread over all collections. It is impossible to tell which collections have more documents matching the query than others, because many collections contain roughly the same numbers of occurrences of the words. The other two queries have the same problem, even though the query words (e.g. "blood", "pressure" and "forest") are less uniformly distributed. Just because "high", "blood" and "pressure" occur frequently in a collection does not mean it necessarily contains any documents matching the query "high blood pressure". The collection could contain many documents about "high air pressure" and "blood donation". The root reason for the above problem is that the document boundary information is lost when the collections are represented as virtual documents in the collection selection index. (Keeping document boundary information results in a collection selection index about the total size of the collections.) We can not tell whether the query words co-occur in the documents or not. As the result, collection selection based on the frequency information of common query words will not work well.

We propose two techniques to address the problem. One technique is to use phrases in the collection selection index. That is, the virtual document for a collection stores not only single words and their frequencies, but also phrases and their frequencies in the collection. Phrases are defined as noun phrases in the documents and queries, and are recognized by a part of speech tagger, JTAG [21]. Including phrases in the collection selection index will at least partially compensate for the loss of document boundary information. Simple queries such as "the White House" and "George Washington" are properly handled. But for longer queries and queries which do not require query words to be adjacent, the loss of document boundary information is still a problem.

The other technique is query expansion. Query expansion has been extensively studied to address the vocabulary mismatch between queries and documents [17, 14, 3, 22]. Our use of query expansion in this paper has a different motivation. As we said before, queries using common words may be adequate for document retrieval but do not provide a good basis for collection selection. We hope that query expansion will add words which are more specific than the words in the original query. The expansion words do not change the underlying information need, but make the expanded query more suitable for collection selection. We hope query expansion will provide some so-called topic words for a query and also increase the mutual disambiguation of common query words. A topic word for an information need is a word which by itself is a very strong indicator of relevance. To illustrate, let us consider the example query "the White House". The words "white" and "house" are ambiguous because they can be used in numerous contexts. But when they co-occur, they are not. If query expansion is based on the whole query "the White House", we will find expansion words such as "Clinton" and "president". Based on our experience, topic words often exist for an information need. For example, for the query "high blood pressure", the topic words may be "cholesterol" and "hypertension". For the query "airplane accident" they may be "NTSB" and "FAA". Topic words are ideal for collection selection because the loss of document boundary information in the collection selection index is a much less serious problem for them.

The query expansion technique used in this paper is local context analysis (LCA) [22]. There are two reasons we choose LCA. Firstly, LCA has shown to be an effective query expansion technique and is completely automatic. Secondly, it requires expansion terms to co-occur with all words in the original query. Therefore it is more likely to pick the topic words for a query than the techniques that expand the query words independently.

4 Experimental Setup and Collection Statistics

We used three sets of collections for evaluations: TREC3, TREC4 and TREC4-TEST. Table 1 lists some statistics for them. The collections in a set were indexed separately to simulate a real-word distributed IR system. To compare the retrieval performance of searching a set of of distributed collections with that of searching a centralized collection, for each set we also created a single centralized index that contains all the documents in that set.

- TREC3: This set is 2.2 Gigabytes large and contains documents from these sources: Wall Street Journal, Associated Press Newswire, Department of Energy abstracts, Federal Register, and Ziff Davis Computer-Select articles. NIST made the documents in TREC3 available in many small files. Each file is from a single source. We created the TREC3 collections by merging the files from the same source into a number of files, each of which is about 20 Megabytes in size. This resulted in 107 collections.

- TREC4: This set is 2.07 Gigabytes large. Documents sources are Wall Street Journal, Associated Press Newswire, Federal Register, Ziff Davis Computer Select articles, San Jose Mercury News articles, and U.S. Patents articles. We created the collections in TREC4 in the same way as we created the TREC3 collections. The number of collections in TREC4 is 100.

- TREC4-TEST: This set consists of 50 of the 100 collections in TREC4. The other 50 collections in TREC4 forms a training collection, TRAIN, which is to be used by local context analysis to expand the TREC4 queries. The purpose is to test the feasibility of expanding a query on a training collection and searching for it on a test set of collections. To make the training collection as representative of the test set as possible, collections from the same source are evenly assigned to TRAIN and TREC4-TEST. For example, 12 collections in TREC4 are of U.S. Patents articles. 6 of them are in TRAIN and the other 6 are in TREC4-TEST.

Indices for the collections are created and searched by INQUERY [2]. So are the collection selection indices. We think that in a realistic environment a typical user can only afford searching a small number of collections. Therefore, we only search 10 collections for a query in our experiments. The general framework for searching a set of distributed collections for a query in our experiments is:

1. Run the query against the collection selection index and return the best 10 collections.

2. Run the query against each of the 10 collections chosen in Step 1 and return a list of the 30 top ranked documents from each collection. Merge the lists of returned documents based on their scores. This results in a ranked list of 300 documents.

Sets of collections	TREC3	TREC4	TREC4-TEST
Number of queries	50	49	49
Raw text size in gigabytes	2.2	2.07	1.03
Number of documents	741,856	567,529	292,639
Mean words per document	260	299	281
Mean relevant documents per query	196	133	65
Number of words	192,684,738	169,682,351	82,508,763
Number of collections	107	100	50
Mean documents per collection	6933	5675	5852
Megabytes per collection	20.6	20.7	20.7

Table 1: Statistics about the sets of collections used for evaluation

Under this general framework, we will consider a number of variations and observe their effect on retrieval effectiveness. The variations are:

- Phrases vs without phrases in the collection selection index. We will examine the impact of adding phrases to the collection selection index on retrieval effectiveness.
- Query expansion vs without query expansion for collection selection, and for document retrieval.
- The effect of varying the size of the local context analysis training collection on retrieval effectiveness.
- The effect of adding different number expansion concepts on retrieval effectiveness.

Searching a fraction of collections will surely cause some relevant documents to be missed. Precision at high recall will suffer. But in our opinion, precision at low recall is far more important, because few users read more than two dozen documents for a search request. Therefore, in this paper we will only consider precision when 5, 10, 15, 20 and 30 documents are read, even though 300 documents are returned for each query.

In addition to precision figures at the above document cut-offs, the t-test [13] is used to decide whether the performance difference between two methods is statistically significant. To decide whether the improvement by method A over method B is significant, the t-test calculates a p_value based on the performance data of A and B. The smaller the p_value, the more significant is the improvement. If the p_value is small enough (p_value $<$ 0.05), we conclude that the improvement is statistically significant.

Two baselines were used. One is the performance of searching a set of distributed collections using the base queries, without query expansion or phrase indexing. Comparison with this baseline tells us the improvement we make by using a certain technique. The other is the performance of searching a single centralized collection using the base queries. Comparison with this one tells us how far we still need to go to achieve the performance we typically get when searching a centralized collection.

5 Baseline Results

Tables 2 and 3 compare the performance of searching a centralized collection and that of searching a set of distributed collections on TREC3 and TREC4, using the base queries. Phrases are not used in the collection selection index. The performance drop due to searching a set of distributed collections in this case is very significant at all document cut-offs on both sets, ranging from 23% to 32.7%.

	1 central index	distri search 107 indices	
5 docs:	0.6440	0.4960	(-23.0)
10 docs:	0.6060	0.4500	(-25.7)
15 docs:	0.5693	0.3987	(-30.0)
20 docs:	0.5440	0.3760	(-30.9)
30 docs:	0.5080	0.3420	(-32.7)

Table 2: Comparing centralized and distributed searching using base queries on TREC3. The collection selection index is word-based.

	1 central index	distri search 100 indices	
5 docs:	0.5510	0.4163	(-24.4)
10 docs:	0.4633	0.3510	(-24.2)
15 docs:	0.4367	0.3116	(-28.6)
20 docs:	0.4061	0.2796	(-31.1)
30 docs:	0.3578	0.2497	(-30.2)

Table 3: Comparing centralized and distributed searching using base queries on TREC4. The collection selection index is word-based.

Callan, Lu and Croft did similar experiments in [4]. They observed that the effectiveness of searching a set of distributed collections was close to that of searching a centralized collection. But their results were obtained on a small number of collections (17). In comparison, the TREC3 and TREC4 sets used in our experiments contain 107 and 100 collections respectively. Therefore our results are more realistic than theirs.

6 Including Phrases in the Collection Selection Index

We now observe the impact on retrieval effectiveness when we add phrase information in the collection selection index. For each collection, we count the number of documents containing a phrase and add the information to the virtual documents. To simplify software design, a phrase is represented as a single token in the collection selection index, just like an ordinary single word. That is, the phrase "information retrieval" is represented as "information-retrieval". Phrases are also recognized (by processing the natural language text of the TREC topics) and added to base queries. That is, the query "information retrieval" will become "information retrieval information-retrieval" after phrase recognition. The modified queries are then run against the collection selection index to return the best collections, as described before. For document retrieval, however, we still use the base queries. Tables 4 and 5 show the impact on retrieval performance when phrases are added to the collection selection index. The results are noticeably better than that of the word-based collection selection index, 12% on average. The t-test indicates that the improvement on TREC3 is statistically significant at cut-offs 15, 20 and 30 (p_value=0.04, 0.02 and 0.02). The improvement on TREC4 is statistically significant at cut-offs 10, 15, 20 and 30 (p_value=0.04, 0.01, 0.01 and 0.02). But the results are still significantly worse than searching the centralized collections. The results show that adding phrases does partially compensate for the loss of document boundary information in the collection selection index. For the TREC3 topic "Term limitations for members of the U.S. Congress", adding the phrase "term-limitation" significantly improved the retrieval performance. This is not surprising considering the words "term" and "limitations" are too generic to be effective for collection selection. The results strongly suggest that phrases help collection selection. This is different from the use of phrases for ranking documents, where the merit of phrases is still debatable.

	distri search word-based	distri search words-and-phrases
5 docs:	0.4960	0.5280 (+ 6.5)
10 docs:	0.4500	0.5040 (+12.0)
15 docs:	0.3987	0.4640 (+16.4)
20 docs:	0.3760	0.4410 (+17.3)
30 docs:	0.3420	0.4047 (+18.3)

	1 central index	distri search words-and-phrases
5 docs:	0.6440	0.5280 (-18.0)
10 docs:	0.6060	0.5040 (-16.8)
15 docs:	0.5693	0.4640 (-18.5)
20 docs:	0.5440	0.4410 (-18.9)
30 docs:	0.5080	0.4047 (-20.3)

Table 4: Comparing word-based collection selection index with words-and-phrases collection selection index (TREC3)

But using phrases for collection selection also introduces problems. One problem is that additional time is needed to recognize the phrases. Another problem is that it significantly increases the size of the collection selection index. Based on our data, a collection selection index with phrases is about 4 times as large as a collection selection index without phrases. Part of the reason for the sharp increase in space cost is due to the large number of infrequent phrases (occurred once or twice). It appears that most of them can be filtered out to cut space costs without affecting searching effectiveness. Even if the extra cost in time and space is acceptable, we still face potential administrative problems when we apply it in a realistic environment. The sites that own the collections may be unwilling to export the phrase information or may use different procedures to recognize them. Despite these problems, we think adding phrases to the collection selection index is still a valuable option because it is a relatively simple procedure and significantly improves retrieval effectiveness.

	distri search word-based	distri search words-and-phrases
5 docs:	0.4163	0.4490 (+ 7.9)
10 docs:	0.3510	0.3816 (+ 8.7)
15 docs:	0.3116	0.3497 (+12.2)
20 docs:	0.2796	0.3143 (+12.4)
30 docs:	0.2497	0.2741 (+ 9.8)

	1 central index	distri search words-and-phrases
5 docs:	0.5510	0.4490 (-18.5)
10 docs:	0.4633	0.3816 (-17.6)
15 docs:	0.4367	0.3497 (-19.9)
20 docs:	0.4061	0.3143 (-22.6)
30 docs:	0.3578	0.2741 (-23.4)

Table 5: Comparing the word-based collection selection index with words-and-phrases collection selection index (TREC4)

7 Query Expansion

We conjectured in Section 3 that the discriminatory power of typical queries for collection selection can be largely enhanced by adding more specific words through query expansion. In this section we describe experiments that tested the conjuncture.

7.1 Local Context Analysis (LCA)

The query expansion technique used in this paper is local context analysis (LCA) [22]. LCA requires a document collection in order to do query expansion. To expand a query, a retrieval system (INQUERY in our case) retrieves a number of top ranked documents or document portions (passages) and presents them to LCA. LCA analyzes the top ranked documents (or passages) and returns a number of concepts for the query. The concepts returned by LCA include single words and phrases (noun phrases). In our experiments, 50 passages were used for a query.

7.2 Experimental Methodology

In the experiments in this section, the collections used for query expansion and the collections to be searched were the same. That is, the LCA concepts for the TREC4 queries were from the centralized TREC4 collection. The same was done to TREC3. While this arrangement is obviously impractical in practice, the goal of the experiments in this section is to confirm the above conjecture concerning the benefit of query expansion for distributed searching. As we will see in Section 8, it is possible to expand a query using a separate training collection and achieve the same effect.

The expansion parameters were based on those in [22]: 70 LCA concepts (including words and phrases) were added to a query, with decreasing weights assigned to the concepts according to the order in which they were returned by LCA. The collection selection indices in the experiments in this section were word-based. Therefore, in the collection selection stage we kept the LCA words but removed the LCA phrases from the expanded queries.

7.3 Query Expansion for Collection Selection

We first report the results of using query expansion in the collection selection stage only. That is, two different versions of a query were used: the expanded query in the collection selection stage and the base query in the document retrieval stage. If query expansion does improve collection selection, we expect that retrieval performance will be better than that of using the base query in both the collection selection and the retrieval stages. Experimental results on TREC3 support this, as shown by Table 6. Retrieval is improved at all document cut-offs. Improvement at higher cut-offs (15, 20 and 30) is around 15%, which is more noticeable than at lower cut-offs. The t-test indicates the improvement is statistically significant at cut-offs 15, 20 and 30 (p_value=0.03, 0.03 and 0.02), but not at cut-offs 5 and 10.

	distri search base query	distri search qry-expan for collection selection
5 docs:	0.4960	0.5320 (+ 7.3)
10 docs:	0.4500	0.4900 (+ 8.9)
15 docs:	0.3987	0.4573 (+14.7)
20 docs:	0.3760	0.4310 (+14.6)
30 docs:	0.3420	0.3933 (+15.0)

Table 6: The effect of using query expansion for collection selection on TREC3. The collection selection index is word-based.

Although the improvement is significant, we think the results do not fully reflect the power of query expansion for collection selection. A query by query analysis shows that a number of queries were hurt because of inconsistence between the collection selection stage and the retrieval stage. To illustrate the problem, let us consider the query "automobile recalls". The expansion words for this query include "Ford", "model", "cars" and so forth. Using the expanded query for collection selection, the selected collections contain many relevant documents that contain the words "recall", "cars", "model" and "Ford" but missing the original query word "automobile". Retrieval using the original query causes these documents to receive a very low score and therefore hurts the retrieval result. We expect that using query expansion in both collection selection and retrieval stages will eliminate this problem and further improve retrieval performance.

7.4 Query Expansion for Collection Selection and Retrieval

As we expected, using query expansion in both collection selection and retrieval stages at the same time does further improve retrieval results, as shown in Tables 7 and 8. Comparing with the distributed baseline (base queries for collection selection and retrieval), the retrieval effectiveness is improved significantly at all document cut-offs on TREC3 and TREC4. The improvement ranges from 23.5% to 41.2%, with an average of 32%. The t-test indicates the improvement is statistically significant at all cut-offs on both sets (p_values are less than 0.01). The results are very close to those of searching the centralized collections. On TREC4, the drop in precision compared with centralized searching is only 2.6% on average. Precision at cut off 10 is even slightly improved (1.7%) over centralized searching. On TREC3, the drop compared with centralized searching is somewhat more noticeable. At document cut-offs 5 and 10, the degradations are 3.7% and 5.3%. The average degradation is 7.9%. This is understandable considering that TREC3 has a higher baseline.

	distri search base-query	distri search qry-expan
5 docs:	0.4960	0.6200 (+25.0)
10 docs:	0.4500	0.5740 (+27.6)
15 docs:	0.3987	0.5227 (+31.1)
20 docs:	0.3760	0.4920 (+30.9)
30 docs:	0.3420	0.4440 (+29.8)

	1 central index	distri search qry-expan
5 docs:	0.6440	0.6200 (- 3.7)
10 docs:	0.6060	0.5740 (- 5.3)
15 docs:	0.5693	0.5227 (- 8.2)
20 docs:	0.5440	0.4920 (- 9.6)
30 docs:	0.5080	0.4440 (-12.6)

Table 7: The effect of using expanded queries for collection selection and retrieval on TREC3. The collection selection index is word-based.

We use some example queries to illustrate why query expansion helps distributed searching. For the TREC4 topic, "what is being done to ensure that Social Security will not go broke?", the words "social", "security" and "broke" are common enough to occur in most collections many number of times. They are not very useful for identifying the best collections. As a result, the performance of distributed searching (without query expansion) is 60% worse than that of centralized searching for this query. Query expansion dramatically improves the performance of this query by 124%, due to the expansion words "pension", "retiree", "budget", "tax", etc. Such words are more specific and more useful than the words in the original query for collection selection. Other cases where query expansion helps include the query "depletion or destruction of the rain forest affected the worlds weather". The expansion words for this query are "greenhouse", "deforestation" and so forth.

	distri search base-query	distri search qry-expan
5 docs:	0.4163	0.5143 (+23.5)
10 docs:	0.3510	0.4714 (+34.3)
15 docs:	0.3116	0.4367 (+40.1)
20 docs:	0.2796	0.3949 (+41.2)
30 docs:	0.2497	0.3395 (+36.0)

	1 central index	distri search qry-expan
5 docs:	0.5510	0.5143 (- 6.7)
10 docs:	0.4633	0.4714 (+ 1.7)
15 docs:	0.4367	0.4367 (+ 0.0)
20 docs:	0.4061	0.3949 (- 2.8)
30 docs:	0.3578	0.3395 (- 5.1)

Table 8: The effect of using expanded queries for collection selection and retrieval on TREC4. The collection selection index is word-based.

	distri search base-query	distri search qry-expan
5 docs:	0.4286	0.5020 (+17.1)
10 docs:	0.3510	0.4469 (+27.3)
15 docs:	0.3034	0.3891 (+28.2)
20 docs:	0.2827	0.3551 (+25.6)
30 docs:	0.2367	0.3041 (+28.5)

	1 central index	distri search qry-expan
5 docs:	0.5184	0.5020 (- 3.2)
10 docs:	0.4367	0.4469 (+ 2.3)
15 docs:	0.3810	0.3891 (+ 2.1)
20 docs:	0.3500	0.3551 (+ 1.5)
30 docs:	0.3075	0.3041 (- 1.1)

Table 9: The effect of using expanded queries for collection selection and retrieval on TREC4-TEST. The collection selection index is word-based.

8 Query Expansion Using a Training Collection

Experiments in the previous section confirmed our conjecture concerning the benefit of query expansion in a distributed searching environment. In this section we propose and evaluate an approach that makes query expansion practical in a distributed searching environment.

The approach is to let a distributed IR system have a dedicated collection of documents solely for the purpose of query expansion. We call the dedicated collection the *training collection*. When a query is posted, the system first expands it by running LCA on the training collection and then searches for it on the actual set of distributed collections. Ideally, we would like the documents in the training collection and those in the actual collections to have similar coverage of subject matters so that LCA can expand the query properly.

We evaluated this approach on TREC4-TEST. The training collection for LCA is TRAIN. Descriptions about TREC4-TEST and TRAIN are in Section 4. The experiments were carried out in the same way as in the previous section, except that query expansion was performed on TRAIN and evaluation was performed on TREC4-TEST.

Table 9 shows the retrieval result. For distributed searching, the performance of the expanded queries is significantly better than that of the base queries. The improvement at the document cut-offs ranges from 17.1% to 28.5%, with an average 25%. The t-test indicates the improvement is statistically significant at all cut-offs (p_value=0.035, 0.00005, 0.00003, 0.00009 and 0.000001). The result is as good as searching the centralized collection. Comparing with centralized searching, there is only a slight degradation at cut-offs 5 and and 30. Precision at cut-offs 10, 15 and 20 is even slightly better than centralized searching.

8.1 Optimizations

In the previous experiments, 70 LCA concepts were used per query. Using so many concepts will significantly slow retrieval. We would like to see the impact when fewer concepts are added to the queries. Table 10 shows the retrieval results when we vary the number of concepts added to a query. The results show that reducing the number of concepts from 70 to 20 does not affect retrieval effectiveness. In fact, using 30 concepts is even slightly better than using 70. But when only 10 concepts are used per query, retrieval performance suffers, by 5.3% on average.

The use of a dedicated training collection solely for query expansion incurs a space cost and we would like to keep it to a minimum. We now examine the impact of using smaller training collections. So instead of using the full TRAIN collection for query expansion, we use 60%, 40% and 20% of the documents in TRAIN for query expansion. Table 11 shows the retrieval results. Comparing with using full TRAIN, there is only a small degradation (3.6%) in effectiveness when 60% and 40% of the documents in TRAIN are used for query expansion. When 20% of TRAIN is used, the degradation becomes significant (about 14%), but the result is still better than that of using the base queries for distributed searching. The results suggest it is possible to cut the size of the training collection without significantly affecting retrieval effectiveness. The exact amount of training data required still remains unknown, because we don't know whether it is a fixed number of megabytes or a certain percentage of the total size of the collections.

9 A Combined Run

Since both phrases and query expansion improve the performance of distributed searching, it is natural to wonder whether combing the two techniques will result in even better performance. Experiments on TREC4 show that this is indeed the case, as shown by Table 12. The result is modestly better, about 4% on average, than using query expansion alone. The t-test indicates the improvement is statistically significant at cut-offs 5, 20 and 30 (p_value=0.03, 0.01 and 0.02). Since the improvement is small, it may not be worthwhile to combine them considering the extra cost in processing time and space.

	70 concepts	50 concepts	30 concepts	20 concepts	10 concepts
5 docs:	0.5020	0.4939 (- 1.6)	0.5184 (+ 3.3)	0.5143 (+ 2.5)	0.4898 (- 2.4)
10 docs:	0.4469	0.4388 (- 1.8)	0.4347 (- 2.7)	0.4224 (- 5.5)	0.4061 (- 9.1)
15 docs:	0.3891	0.3946 (+ 1.4)	0.3932 (+ 1.1)	0.3905 (+ 0.4)	0.3687 (- 5.2)
20 docs:	0.3551	0.3541 (- 0.3)	0.3571 (+ 0.6)	0.3520 (- 0.9)	0.3347 (- 5.7)
30 docs:	0.3041	0.3027 (- 0.5)	0.3075 (+ 1.1)	0.2986 (- 1.8)	0.2912 (- 4.2)

Table 10: The effect of query expansion size on retrieval performance on TREC4-TEST. The collection selection index is word-based.

	TRAIN	60% TRAIN	40% TRAIN	20% TRAIN
5 docs:	0.5184	0.5061 (- 2.4)	0.5061 (- 2.4)	0.4286 (-17.3)
10 docs:	0.4347	0.4245 (- 2.3)	0.4224 (- 2.8)	0.3694 (-15.0)
15 docs:	0.3932	0.3850 (- 2.1)	0.3741 (- 4.9)	0.3388 (-13.8)
20 docs:	0.3571	0.3347 (- 6.3)	0.3439 (- 3.7)	0.3194 (-10.6)
30 docs:	0.3075	0.2918 (- 5.1)	0.2952 (- 4.0)	0.2646 (-14.0)

Table 11: The effect of the training collection size on retrieval performance on TREC4-TEST. 30 expansion concepts are used per query. The collection selection index is word-based.

	distri search qry-expan word-based for collection selection	distri search qry-expan words-and-phrases for collection selection
5 docs:	0.5143	0.5429 (+ 5.6)
10 docs:	0.4714	0.4878 (+ 3.5)
15 docs:	0.4367	0.4463 (+ 2.2)
20 docs:	0.3949	0.4102 (+ 3.9)
30 docs:	0.3395	0.3517 (+ 3.6)

Table 12: Combining query expansion and words-and-phrases collection selection index vs query expansion alone (TREC4)

10 Conclusions and Future Work

The experimental results show that for typical queries, searching a large set of distributed collections is significantly worse than that of searching a centralized collection. The main reason is that typical queries are not suitable for collection selection. We have proposed two techniques to address the problem. One technique is to use phrase information in the collection selection index. The other is query expansion. Both techniques significantly improve retrieval effectiveness in a realistic distributed search environment. Query expansion, in particular, can make the performance of searching a set of distributed collections close to that of searching a centralized collection.

There are a number of areas in which we will continue our work. One is to test our techniques on even larger collections. One such collection we plan to use is the 20 Gigabytes TREC VLC (Very Large Corpus) collection. We plan to break it into around 1000 collections. This will make our test environment even more realistic. Another area is to find a "versatile" training collection for query expansion. Such a collection should have a wide coverage of subject matters so that most queries can be properly expanded. One possible choice is to pull out all the newspaper articles (Associated Press, Wall Street Journal, LA Times, etc) from the TREC collections to form a training collection. The third area is user interaction. Query expansion occasionally hurts a query by adding bad terms. We would like the user to control what terms to be ultimately used to expand his/her query. Our experiments show that even 20 concepts per query can make a big difference in retrieval performance. This means that the amount of work to manually edit the expanded query is reasonable. Finally, our technique for merging retrieval results needs to be improved. It merges documents from different collections based on locally computed scores. As pointed out in [4], this simple technique has certain drawbacks. How to effectively merge retrieval results from different collections is an active research topic by itself. Any progress in that area will benefit us as well.

Acknowledgements Thanks to Bruce Croft for his guidance throughout this study. Thanks to Hongmin Shu for her help in doing the experiments.

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,and also supported in part by United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor(s).

References

[1] C. Baumgarten. A probabilistic model for distributed information retrieval. In *Proceedings of the*

20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 258–266, 1997.

[2] J. Broglio, J. P. Callan, and W.B. Croft. An overview of the INQUERY system as used for the TIPSTER project. In *Proceedings of the TIPSTER Workshop*. Morgan Kaufmann, 1994.

[3] Chris Buckley, Amit Singhal, Mandar Mitra, and Gerard Salton. New retrieval approaches using SMART : TREC 4. In *Proceedings of the TREC 4 Conference*, 1996.

[4] J. P. Callan, Z. Lu, and W.B. Croft. Searching distributed collections with inference networks. In *Proceedings of 18th ACM SIGIR International Conference on Research and Development in Information Retrieval*, pages 21–28, 1995.

[5] J.P. Callan, W.B. Croft, and J. Broglio. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, pages 327–343, 1995.

[6] Anil Chakravarthy and Kenneth Hasse. NetSerf: Using semantic knowledge to find Internet information archives. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1995.

[7] Alice Chamis. Selection of online databases using switching vocabularies. *Journal of the American Society for Information Science*, 39(3), 1988.

[8] Norbert Fuhr. A decision-theoretic approach to database selection in networked IR. Technical report, Computer Science Department, University of Dortmund, 1996.

[9] L. Gravano, Kevin Chang, H. García-Molina, and Andreas Paepcke. STARTS Stanford protocol proposal for Internet retrieval and search. Technical report, Computer Science Department, Stanford University, 1996.

[10] L. Gravano and H. García-Molina. Generalizing GlOSS to vector-space databases and broker hierarchies. In *Proceedings of the 21th VLDB Conference*, 1995.

[11] L. Gravano, H. García-Molina, and A. Tomasic. The effectiveness of GlOSS for the text database discovery problem. In *Proceedings of SIGMOD 94*, pages 126–137. ACM, September 1994.

[12] D. Harman. Overview of the Third Text REtrieval Conference (TREC-3). In D. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 1–20. NIST Special Publication 500-225, 1995.

[13] David Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–338, 1993.

[14] Y. Jing and W. Bruce Croft. An association thesaurus for information retrieval. In *Proceedings of RIAO 94*, pages 146–160, 1994.

[15] Brewster Kahle and Art Medlar. An information system for corporate users: Wide Area Information Servers. Technical Report TMC199, Thinking Machines Corporation, 1991.

[16] Alistair Moffat and Justin Zobel. Information retrieval systems for large document collections. In D. Harman, editor, *The TREC3 Proceedings*, 1995.

[17] Karen Sparck Jones. *Automatic Keyword Classification for Information Retrieval*. Butterworth, London, 1971.

[18] Howard R. Turtle. *Inference Networks for Document Retrieval*. PhD thesis, University of Massachusetts at Amherst, 1990.

[19] Charles Viles and James French. Dissemination of collection wide information in a distributed information retrieval system. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 12–20, 1995.

[20] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. The collection fusion problem. In D. K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, 1995. National Institute of Standards and Technology, Special Publication 500-225.

[21] Jinxi Xu, John Broglio, and Bruce Croft. The design and implementation of a part of speech tagger for English. Technical Report IR52, CIIR, Computer and Information Science Department, University of Massachusetts, Amherst, MA 01003, 1994.

[22] Jinxi Xu and Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, 1996.