Collection Selection and Results Merging with Topically Organized U.S. Patents and TREC Data

Leah S. Larkey, Margaret E. Connell
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{ larkey, connell } @cs.umass.edu

Jamie Callan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
callan@cs.cmu.edu

ABSTRACT

We investigate three issues in distributed information retrieval, considering both TREC data and U.S. Patents: (1) topical organization of large text collections, (2) collection ranking and selection with topically organized collections (3) results merging, particularly document score normalization, with topically organized collections. We find that it is better to organize collections topically, and that topical collections can be well ranked using either INQUERY's CORI algorithm, or the Kullback-Leibler divergence (KL), but KL is far worse than CORI for non-topically organized collections. For results merging, collections organized by topic require global *idf*s for the best performance. Contrary to results found elsewhere, normalized scores are not as good as global *idf*s for merging when the collections are topically organized.

Keywords

Information retrieval, Collection selection, Topical organization.

1. INTRODUCTION

We have developed a distributed system for the search and classification of U.S. patents [11], using INQUERY, a search engine developed at the Center for Intelligent Information Retrieval at the University of Massachusetts [3]. Our design choices were guided by recent research on managing large text collections and retrieving documents from distributed databases. The performance of our system led us to question the applicability of these methods to collections organized by topic, and stimulated the present research.

Most research on searching distributed collections has focused upon two issues (1) *Collection ranking*: ranking collections and selecting from them a small number to search for a given query, and (2) *Results merging*: combining the ranked lists of documents returned from each of the selected collections into a single ranked list. Our research addresses these and a third important issue, that of (3) *topical organization*: the subdivision of data by topic, and its interaction with collection ranking and results merging.

CIKM 2000, November 7-10, 2000, Washington, D.C.

The advantages of dividing a very large text collection into smaller collections include faster response time, simplification of administration, and the possibility of restricting the search to the best part of the collection. The obvious disadvantage is that one cannot retrieve documents from collections outside of the selected, top-ranked set. In spite of this disadvantage, some recent studies have claimed that given good organization of data, collection ranking, and results list merging, one can achieve retrieval performance from distributed databases that approaches that from a single centralized database [12] [19].

We investigate how best to organize data by comparing retrieval from collections organized topically with retrieval from collections organized chronologically or by source, using TREC data and U.S. Patent data. We investigate the second issue, collection ranking for topically organized collections, by comparing two collection ranking algorithms on the TREC and patent collections. Third, we address results merging for topically organized collections by comparing four different merging algorithms on patent and TREC collections under topical and non-topical organizations.

This is the first collection selection study involving large data sets that supplements TREC data with another collection. This is important to avoid bias. Our research is the first to examine retrieval from topically organized collections that are not subdivided by clustering, but by a human-designed category scheme of considerable abstractness and complexity. Our investigation of different merging algorithms with topically organized data is also unique.

2. PREVIOUS RESEARCH

2.1 Topical Organization

We look at three ways of subdividing large corpora: by date, by source, and by topic. Chronological organization is particularly appropriate for corpora with a continual influx of new documents, such as news archives or patents. A new collection can be added for each week, month, year, etc. Chronologically organized sets of collections tend to have convenient statistical properties, such as similar sizes and term frequency distributions. The disadvantage is that documents relevant to a query may be scattered throughout the collections, allowing little chance of finding them in a search restricted to a small number of collections, unless the query concerns something like a news event which gets most of its coverage in a narrow time window.

The second common mode of organization is by source, for example, Associated Press, Wall Street Journal, Federal Register, etc., which can simulate retrieval from different providers. Organization by source falls between topical and chronological organization in that different sources tend to concentrate on somewhat different content.

Under topical organization, documents about similar subjects are grouped into the same collection. If this grouping is well done, most or all of the relevant documents for a query should potentially be found in one or a small number of collections, according to van Rijsbergen's cluster hypothesis that closely associated documents should be relevant to the same queries [14]. In the tradition of early work on clustering documents and evaluating queries against clusters rather than single documents [7][9], Xu and Croft [19] have shown that for TREC queries, topical organization by global clustering does in fact concentrate most relevant documents into a small number of collections. Xu and Croft divided TREC collections into 100 subcollections either by source or by topic using clustering. They found far better retrieval performance with subcollections divided by topic compared to the heterogeneous subcollections divided by source. Retrieval from the best 10 topical subcollections was comparable to centralized retrieval, whereas the retrieval from the 10 best source-based subcollections showed 25-30% lower precision than centralized retrieval.

In creating our distributed patent system, we chose a topical organization by patent class because each U.S. patent belongs to one of 400 patent classes. Unlike the TREC clusters, patent classes are of human design, and are currently in active use by patent searchers and the USPTO (United States Patent and Trademark Office). Patents have been manually assigned to the classes according to extremely abstract criteria. Automatic classification into patent classes works surprisingly poorly, suggesting that these groupings are not what one would obtain by clustering. These data provide a good testbed for generalizing the clustering results to a topical organization with an extremely different basis.

2.2 Collection Ranking

Most collection selection research considers distributed collections that are autonomous and private. It is assumed to be too costly to query all the available collections, so a small number must be selected. Some researchers rely on manually created characterizations of the collections [4], others require a set of reference queries or topics with relevance judgements, and select those collections with the largest numbers of relevant documents for topics that are similar to the new query [17].

We are interested in the class of approaches including CORI [1], gGlOSS [6], and others [8][20], that characterize different collections using collection statistics like term frequencies. These statistics, which are used to select or rank the available collections' relevance to a query, are usually assumed to be available from cooperative providers. Alternatively, statistics can be approximated by sampling uncooperative providers with a set of queries [2]. In the present study we compare two of these approaches, CORI and topic modeling.

The distributed patent system uses the CORI net (collection retrieval information network) approach in INQUERY [1], described in more detail in section 3.3.1, because this method has been shown successful in ranking collections, and outperforms some of the best alternative approaches [5]. Given our topically organized data, we thought we might get better performance from the topic modeling approach used by Xu and Croft [19] to rank their clustered collections.

A topic model is a probability distribution over the items in a corpus, in this case unigrams. The Kullback-Leibler (KL) divergence [10], an information theoretic metric used to measure how well one probability distribution predicts another, was applied to measure how well a topic model predicts a query or a document. In Xu and Croft, this topic modeling approach performed better than CORI on clustered TREC4 data according to two measures: higher ranking collections contained larger numbers of relevant documents, and retrieval attained higher precision. It is significant, however, that the same KL measure was used in creating the topical clusters. This method of collection selection may be uniquely suited to selection when the collections have been organized based on the same KL metric. Xu and Croft's results leave open two issues we address here, (1) whether KL is superior to CORI even when the topical scheme is not tied so closely to the retrieval metric, and (2) how topic modeling performs when collections are not organized according to topics.

2.3 Results Merging

In a typical distributed retrieval situation, document scores from different providers may be computed differently, or not be provided at all. To present the user with a single ranked list, these lists must be merged into an accurate single ordering. When no scores are provided, solutions depend only on the ranking of collections and the number and ordering of documents retrieved from each collection [16]. When scores are provided, one can attempt to scale the disparate scores [15][18]. Even in our relatively consistent situation where all the document scores are provided by INQUERY, the differences in the statistical makeup of the collections present a barrier to an accurate ordering of documents from different collections. In the typical *tf·idf* document score computed in INQUERY and most other systems [3][13], the *idf* (inverse document frequency) component is a function of the number of documents in the collection containing the query terms, so that identical documents in different collections would receive different document scores.

One approach, taken by Xu and Croft [19], is to avoid the problem by using *global idf*s, i.e. *idf*'s from the full set of documents in all the collections, in computing document scores. In INQUERY, we compute normalized document scores which are scaled using maximum and minimum possible scores to attempt to make them comparable across collections. Powell et al. [12] found that TREC document scores could be effectively normalized this way, yielding retrieval performance as good as that attained via global *idf*. However, the document rankings we obtained from our distributed PTO system suggested that this normalization was not sufficient for patent data.

When we searched the distributed patent database, we would often find apparently non-relevant documents at the top of the list and good documents at lower ranks. In contrast, when we searched a single database containing two years of patents, we would get good retrieval results. A closer analysis of the situation revealed that the collection ranking algorithms were doing a good job of selecting collections, but that documents from lower-ranking collections (among the top 10) were outranking documents from higher ranking collections. Thus the problem was one of results merging.

Table 1 shows one example of such a pattern, obtained from the query "Accordion musical instrument." The ranked list of patent classes for the query is above, and the ranked list of documents after merging is below. The number to the left of each document title indicates the patent class, and hence, the collection, where the document resides. Merging was based on INQUERY's normalized document scores.

Class	Class Description
084	Music
381	Signal Processing Systems and Devices
181	Acoustics
446	Amusement devices: Toys
434	Education and demonstration
281	Books, strips, & leaves
369	Dynamic information storage or retrieval

	Patent Title
369	Automatic musical instrument playback from digital source
369	Electronic apparatus with magnetic recording device
369	Method and apparatus for restoring aged sound recordings
369	Auto-playing apparatus
369	Disc playing apparatus …
369	Subcode info and block ID system for a disc player
381	Microphone pickup system
084	Slender housing for electronic M.I.D.I. accordion
084	Accordion support apparatus
084	Electronic accordion housing and support stand
084	Accordion with new order of sounds

Table 1. Problem Query Example. Ranked list of classes and patents for query "Accordion Musical Instrument".

In this example, many patents that mention music, instruments, and accordions, are in the best class for the query, *music*. In this class each of these query terms has a relatively low *idf*. For less relevant collections, these terms are rare, and hence have higher *idf*s, which results in higher document scores for the documents in the lower-ranked collections. Normalization should compensate for the disparity, but was not fully successful. This rare term problem has been noted before [16][19]. However, in the PTO situation, the rare term problem is not at all rare. Due to the skewed term distributions across collections and the short, specific PTO queries, most query terms are rare terms.

The failure in the PTO system of normalization methods that were successful in other distributed systems motivated the merging part of our research. There is no prior research on merging and normalization methods for topically organized collections. We compare several different merging algorithms, with TREC data and with patent data, organized topically and otherwise.

3. EXPERIMENTAL METHOD

We use two different data sets in this research, which we refer to as TREC3 and PTO. Their statistics can be seen in Table 2

3.1 TREC Data

The TREC3 data set is the TREC3 data set reported in Xu and Croft [19]. This set of 741,856 documents was broken up into 100 collections in two ways, *by topic* and *by source*. The by-topic organization was Xu and Croft's TREC3-100col-global set. The documents were clustered by a two pass K-means algorithm using the Kullback-Leibler divergence as the distance metric. The by-source organization was Xu and Croft's TREC3-100col-bysource set. Here, the documents were grouped by source, allocating a number of collections to each source that was proportional to the total number of documents from that source. The 50 TREC3 queries were based on TREC topics 151-200.

3.2 PTO Data

The PTO data set is made up of virtually all utility and plant patents from the years 1980 through 1996, which number around 1.4 million. This is about one fourth of all U.S. utility and plant patents, and comprises 55 megabytes of text. We excluded design patents, because the content of a design claim is usually an image rather than a text description. Patents range in size from a few kilobytes to around 1.5 megabytes. We include the full text of all of these patents in our collections.

The set has been divided into subcollections in two different ways for this research. The *chrono* set is divided chronologically into 401 collections of roughly equal size in terms of numbers of patents. The *by-class* set is divided by patent class into 401 subcollections.

There is no standard set of patent queries with relevance judgments for the patent collection. We constructed 37 queries covering a range of patent areas, non-technical enough for laymen to consistently judge the relevance of patents to queries. We had searched the patent collection at various times in the past to look for prior art, and some of the queries came from these searches.

Data Set	Size		Avg. Doc Len	Collections	Docs per Collection		
	GB	Num Docs	Words	Number	Avg	Min	Max
PTO by class	55	1,397,860	5586	401	3486	1	34,271
PTO chrono	55	1,397,860	5586	401	3486	3,461	3,486
TREC3 by topic	2.2	741,856	260	100	7418	100	106,782
TREC3 by source	2.2	741,856	260	100	7418	7,294	7,637

Table 2: Test Collection Summary Statistics

Data Set	Num Queries	Words per Query			Rel Docs per Query		
		Avg	Min	Max	Avg	Min	Max
PTO	37	3.0	1	7	35	9	68
TREC3	50	34.5	15	58	196	14	1141

Table 3: Query Summary Statistics

Two of the three experimenters judged the relevance of documents to these queries. We collected the top 30 documents returned for each query pooled over all the experimental conditions. This total pool of documents for a given query was judged by a single experimenter for consistency, in a random order so the judge would be unaware of which condition(s) retrieved the document. Because there was a great deal of overlap in the sets of documents retrieved for a query across the different conditions, an average of 90 documents were judged per query. Table 3 shows more information about the queries.

3.3 Distributed Retrieval

Retrieval consisted of the following steps in all experimental conditions:

(1) Rank the collections against the query. The collection ranking methods are either CORI or KL, described below.

(2) Retrieve the best 30 (for PTO) or 100 (for TREC) documents from each of the ten top ranked collections, using the same algorithm as in INQUERY's single collection retrieval system [3], modified to make available the maximum and minimum possible document scores for normalization.

(3) Normalize scores, if appropriate to the experimental condition, and merge the results lists. The four merging methods are described in detail below. The baseline method is *global idf*, and other three conditions are normalization techniques we call *norm-both*, *norm-dbs*, and *norm-docs*. For TREC, we also provide a centralized retrieval baseline, in which documents are retrieved from a single large database.

To address the topical organization issue, we query the patent collections organized by class and by date, and the TREC collections organized by topic and by source.

To evaluate retrieval we look at precision at 5, 10, 15, 20, and 30 (and 100 for TREC) documents. We use this measure rather than the more usual 11 point precision, because of the relatively small number of relevant documents we have for the PTO queries.

3.3.1 CORI Collection Ranking

In the CORI net approach, collection ranking is considered to be analogous to document ranking. Collections are treated as pseudo-documents, and ranked according to the following analogue to *tf·idf* scores for document retrieval from single collections [1]. This formulation, for a simple "natural language" query with no special operators, is as follows:

$$Score_C = \frac{1}{|Q|} \cdot \sum_{j=1}^{|Q|} \left(.4 + .6 \cdot T_j \cdot I_j\right)$$

$|Q|$ is the number of terms in the query, Tj is the *tf* analogue for term *j*, that is:

$$T_j = \frac{df_j}{df_j + 50 + 150 \cdot \left(cw / avg\_cw\right)}$$

and I_j is the *idf* analogue for term *j,* that is:

$$I_j = \frac{\log\left(\left(N + 0.5\right) / cf_j\right)}{\log\left(N + 1.0\right)}$$

where df_j is the number of documents in collection C containing the j^{th} query term, *cw* is the number of indexing terms in C, *avg_cw* is the average number of indexing terms in each collection, N is the number of collections, and cf_j is the number of collections containing term j.

3.3.2 KL Collection Ranking

In Xu and Croft's language modeling approach, collections are ranked by a modification of the Kullback-Leibler (KL) divergence which measures the distance between a query Q and a collection C:

$$Score_C = \sum_{j=1}^{|Q|} \frac{f(Q, w_j)}{|Q|} \log \frac{f(Q, w_j) / |Q|}{\left(f(C, w_j) + f(Q, w_j)\right) / \left(|Q| + |C|\right)}$$

where $f(Q, w_j)$ is the number of occurrences of term w_j in the query, $|Q|$ is the number of term occurrences in the query, $f(C,w_j)$ is the number of occurrences of the term w_j in the collection, and $|C|$ is the total number term occurrences in the collection.

3.3.3 Normalization for Merging

In INQUERY, document scores are normalized based on the maximum (D_{max}) and minimum (D_{min}) scores any document could attain for the query: $D_{norm} = \left(D - D_{\min}\right) / \left(D_{\max} - D_{\min}\right)$.

Collection scores are similarly normalized using the maximum (C_{max}) and minimum (C_{min}) scores a collection could attain for the query: $C_{norm} = \left(C - C_{\min}\right) / \left(C_{\max} - C_{\min}\right)$.

The final ranking score for a document combines the normalized collection and document scores into a final score for the document which we call *norm-both*, because both document and collection scores are normalized:

norm-both: $Score = \left(D_{norm} + 0.4 \cdot C_{norm} \cdot D_{norm}\right) / 1.4$

Two other normalization methods are variations of the norm-both approach. *Norm-docs* simply uses the normalized document score, without considering any contribution of collection scores. *Norm-dbs* combines the raw document score with a normalized collection score.

norm-docs: $Score = D_{norm}$

norm-dbs: $Score = (D + 0.4 \cdot C_{norm} \cdot D) / 1.4$

Norm-dbs was of interest because it was the method in use when we first noticed the rare term problem described above. It is the only one of these three normalization methods that requires only a list of documents and scores. The other methods require ideal maximum and minimum scores for each query, which would not be available from an uncooperative provider. Norm-docs was included under the reasoning that perfect normalization should yield scores similar to global *idf*.

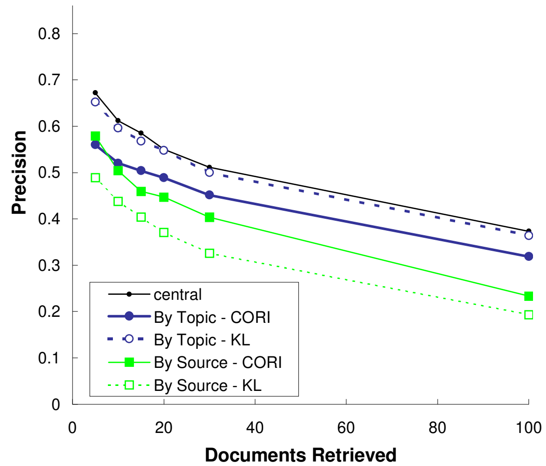

Figure 1. TREC precision, CORI vs KL collection ranking, organization by topic and by source.

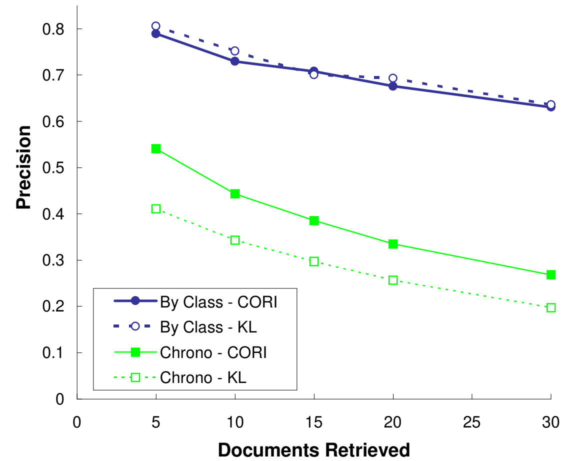

Figure 2. PTO precision, CORI vs KL collection ranking, organization by class and chronological.

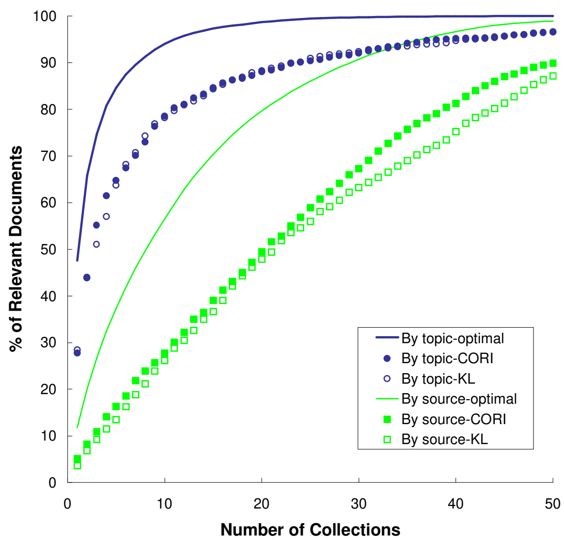

Figure 3. TREC Distribution of relevant documents in top 50 collections, organization by topic and by source, CORI vs KL ranking

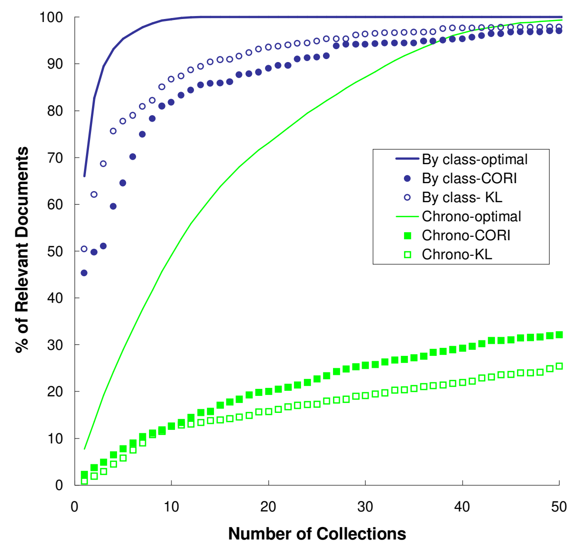

Figure 4. PTO Distribution of relevant documents in top 50 collections, organization by class and chronological, CORI vs KL ranking

4. RESULTS

4.1 Topical Organization

Figure 1 shows that for TREC data, organization by topic gives better retrieval results than organization by source, replicating Xu and Croft's findings for KL collection ranking (open symbols and dotted lines) and extending these findings to CORI collection ranking (filled symbols and solid lines). As anticipated, the larger PTO data set also shows this pattern (Figure 2). This topical superiority holds for all other methods of result list merging, as we will illustrate below.

One reason for the topical superiority can be seen in Figure 3 and Figure 4, which show the distribution of relevant documents in the top 50 collections as ranked by the CORI and KL algorithms. The optimal curves represent the case where the collections are ordered by the actual number of relevant documents in each, averaged over all queries. This provides an upper bound for collection ranking algorithms. When collections are organized by topic (circles in the plots), relevant documents tend to be concentrated into a small number of collections. When collections are not organized by topic (squares), relevant documents are more scattered throughout collections, limiting the number of documents that can be retrieved from 10 collections.

Interestingly, the advantage for topical organization is much more pronounced for the PTO data than for the TREC data. This appears to be both because topical organization is better for PTO than for TREC and because the non-topical organization is worse for PTO than for TREC. Relevant documents are more concentrated into a smaller proportion of PTO by-class collections than the TREC by-topic collections: The top 10 PTO collections are only 2.5% of the 400 total collections, but cover 83.7% of the known relevant documents. The top 10 TREC collections cover 10% of the data but include only 78.5% of the known relevant documents. On the other hand, chronological organization for PTO is worse than organization by source for TREC, in that relevant PTO documents are more evenly spread across collections.

4.2 Collection Ranking Methods

The same figures illustrate the comparison of collection ranking algorithms, the Kullback-Leibler divergence and INQUERY's CORI algorithm, addressing the generality of the claim that KL is a better way to select topically organized collections. Collections were ranked either via CORI or KL. We consider only global *idf* here to separate the collection ranking issue from that of merging.

We replicated Xu and Croft's findings that KL yields better retrieval performance than CORI on topically organized TREC data (Figure 1). KL retrieval is almost as good as retrieval from a single centralized collection. However, KL is better than CORI only on topically organized data. KL performs worse than CORI on TREC data organized by source.

On the PTO data organized by class in Figure 2, the KL metric shows only a very small advantage over CORI, if any. Compared to the large KL advantage on TREC, the KL advantage on topical PTO data is very small. KL performs substantially worse on the non-topical PTO data than does CORI.

The corresponding distributions of relevant documents across collections as ranked by KL and CORI (Figure 4 and Figure 3) show that there is not much difference between KL and CORI in the number of relevant documents seen in the first 10 collections. This lack of a difference holds for PTO and TREC data, and in the topical and non-topical conditions. However, if we retrieved documents from more than 10 collections, we would have seen differences between CORI and KL in the numbers of relevant documents available.

The distributions of relevant documents across collections in Figure 3 and Figure 4 are difficult to interpret. For both organizations, by topic and by source, the distributions show essentially the same proportion of relevant documents in the top ranking 10 collections, whether they are ranked by CORI or by KL. We cannot attribute the better performance of KL on topically organized data to its choosing collections with more relevant documents. Instead, KL somehow selects collections where the relevant documents receive higher INQUERY scores. Similarly, on the TREC data organized by source, KL selects collections with about the same number of relevant documents as CORI, but these documents receive lower scores, and hence lower ranks.

4.3 Document List Merging

The picture is also complicated when we consider document list merging. For the topical PTO data in Figure 5 we see large differences in precision between merging algorithms. Global *idf* is better than norm-both, which is better than norm-docs, which is better than norm-dbs. When the PTO data are organized chronologically, all the merging techniques yield the same precision (Figure 6). This lack of difference is due to the fact that all the chronological subcollections have very similar term statistics. Therefore, document scores from single collection retrieval are already normalized relative to each other, and further normalization makes no difference.

The TREC results show much smaller differences among merging algorithms than the PTO results show. When the organization is by topic, (Figure 7), global *idf* is better than all three normalization methods, which are indistinguishable from one another. When the organization is by source (Figure 8), global *idf* is only slightly better than the other merging methods. In contrast to the findings of Powell, et al.[12], we find that global *idf* gives better results than any normalization.

Taken together, the PTO and TREC results show that for topically organized data, global *idf* is preferable to any of the normalization methods above. This result is contrary to the claims of Powell, et al. that by normalizing both document and collection scores one can attain merging performance that is as good as using global *idf*. The key factor is probably the degree of skew in the term frequency distributions of the different collections. The PTO division by class is extreme in that term frequencies for a query word can vary greatly in different subcollections, so that documents from different subcollections can have extremely disparate scores. Normalization is not sufficient to overcome the skewed scores for PTO. However, it can compensate for the differences among less skewed subdivisions

5. DISCUSSION

5.1 Topical Organization

We have shown superior retrieval from collections that are subdivided along topical lines. Division of patents by chronology, in contrast, produces subcollections that cannot be distinguished from one another statistically, and can therefore not be effectively ranked by any selection algorithm. A TREC3 subdivision by source falls between a topical organization and a chronological organization. With division by source, similar documents are somewhat concentrated into subcollections, and hence there is potential for retrieval from a small number of collections to be effective.

In our experiments, topical organization seemed to have a larger effect with PTO data than with the TREC data, perhaps because of the comparison to the chronological baseline, which is less organized than TREC's by-source baseline. There is more going on, however. The distributions seem to show more concentration of relevant documents into fewer subcollections for PTO by class than for TREC by topic. It is possible, however, that this is an artifact of our judging only documents that were retrieved in our experiments, or of the queries being particularly aimed at one or a small number of patent classes. Or it may be that the existing manual patent classification system is a better organization for patent searching than global clustering is for TREC queries.

.

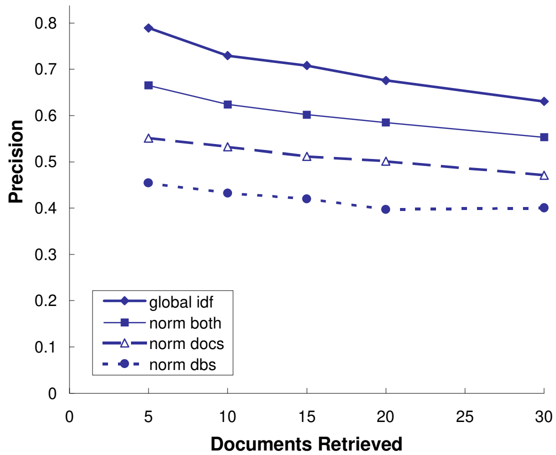

Figure 5. PTO by class precision for four results merging algorithms

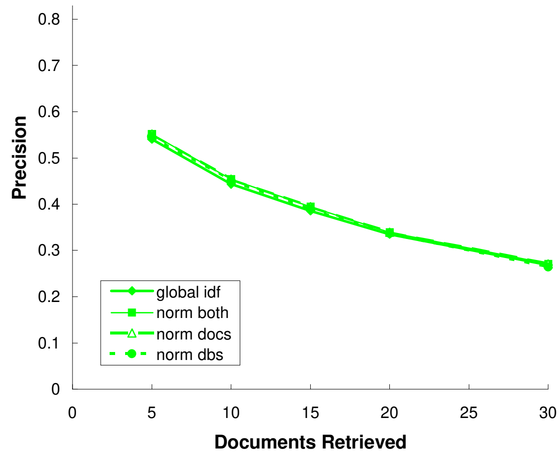

Figure 6. PTO chrono precision for four results merging algorithms

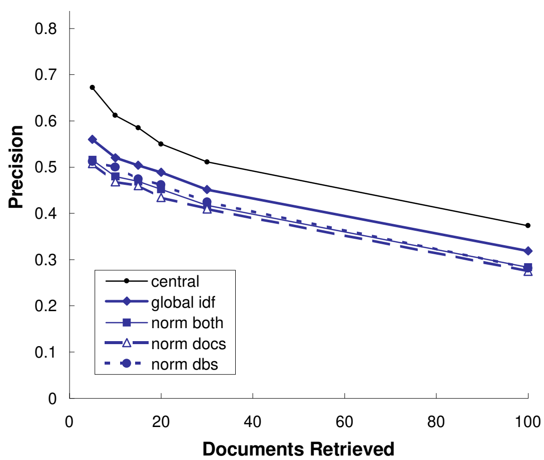

Figure 7. TREC by topic precision for results merging algorithms

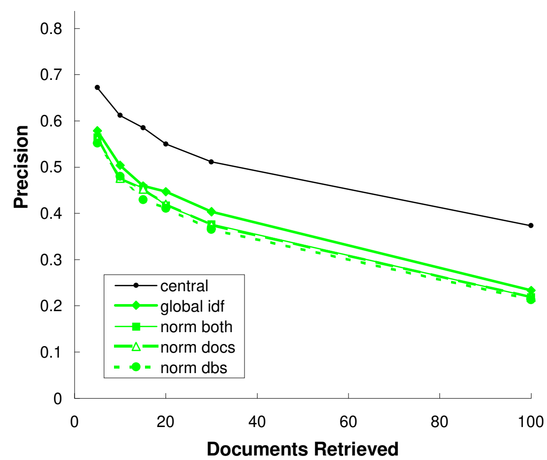

Figure 8. TREC by source precision for results merging algorithms

5.2 Collection Ranking

The comparison of CORI with KL collection ranking methods confirmed that KL is clearly better than CORI when the subcollections have been clustered using KL. On PTO data, where the topics are based on human-designed classes, KL shows only a very small gain, if any, over CORI in the distribution of relevant documents and no gain in precision. However, KL gives worse results than CORI when collections are not organized by topic, as we see with the TREC by-source results and with the PTO chronological results. KL is effective for topical organizations, but should not be used when collections are not organized topically.

5.3 Results Merging

We have shown that for results merging, none of the three normalization methods works as well as global *idf*, for both PTO and TREC data sets. We found big differences among the normalization methods on the PTO data. It is more effective to normalize both collection and document scores and combine them, than it is to normalize either scores alone. However, in contrast to Powell's results, none of these versions of normalization perform as well as using global *idf*, probably because the term distributions are so skewed.

5.4 Implications

The results of this study suggest that the best way to implement the distributed patent search system is to divide up the collection by patent class, to use CORI or KL for collection ranking, and to use global *idf* for merging.

This pattern of results has some bearing upon how one might want to merge results lists in the case of retrieval from disparate providers when one cannot control (or even know) how document scores are computed, or in the worse case, when the provider returns no document scores at all. One could compute INQUERY style document scores for the top *n* documents on each results list using just the text of the documents and the collection wide frequency information available in the collection-selection database, which was either obtained by cooperation from providers or estimated by sampling. The *tf* part of the *tf·idf* score could be derived by parsing the documents and counting occurrences of query words in the documents. The *idf* component is a simple function of the frequency information in the collection. It would require very high bandwidth to get the text of all the documents to be ranked, but as connections get faster this will be possible.

6. ACKNOWLEDGMENTS

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 authors' and do not necessarily reflect those of the sponsors.

7. REFERENCES

[1] Callan, J.P., Lu, Z., and Croft, W. B. Searching Distributed Collections with Inference Networks. In *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21-28, 1995.

[2] Callan, J., Connell, M., and Du, A. Automatic Discovery of language models for text databases. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, pages 479-490, 1999.

[3] Callan, J. P, Croft, W. B., and Broglio, J. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31(3), pages 327-343, 1995.

[4] Chakravarthy, A. S. and Haase, K. B. Netserf: Using Semantic Knowledge to Find Internet Information Archives. In SIGIR '95: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4-11, 1995.

[5] French, J. C., Powell, A. L., Callan, J., Viles, C. L., Emmitt, T., Prey, K. J., and Mou, Y.. Comparing the Performance of Database Selection Algorithms. In *Proceedings of SIGIR '99: 22nd International Conference on Research and Development in Information Retrieval*, pages 238-245, 1999.

[6] Gravano, L., and Garcia-Molina, H. Generalizing GLOSS to vector-space databases and broker hierarchies. In *Proceedings of the 21st International Conference on Very Large Databases (VLDB)*, pages 78-89, 1995.

[7] Griffiths, A., Luckhurst, H. C., and Willet, P. Using Interdocument Similarity Information in Document Retrieval Systems. *Journal of the American Society for Information Science*, 37, pages 3-11, 1986.

[8] Hawking, D., and Thistlewaite, P. Methods for Information Server Selection. *ACM Transactions on Information Systems*, 17(1), pages 40-76, 1999.

[9] Jardine, N. and van Rijsbergen, C.J. The use of hierarchical clustering in information retrieval. *Information Storage and Retrieval*, 7, pages 217-240, 1971.

[10] Kullback, S. Keegel, J. C., and Kullback J.H. *Topics in Statistical Information Theory*. Springer Verlag, 1987.

[11] Larkey, L. S. A Patent Search and Classification System. In *Digital Libraries 99, The Fourth ACM Conference on Digital Libraries*, pages 79-87, 1999.

[12] Powell, A. L., French, J. C., Callan, J., Connell, M., and Viles, C. L. The Impact of Database Selection on Distributed Searching. In *SIGIR 2000: Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232-239, 2000.

[13] Salton, G. *Automatic Text Processing*. Addison-Wesley, 1989.

[14] van Rijsbergen, C.J. *Information Retrieval*. Butterworths, second edition, 1979.

[15] Viles, C. L. and French, J. C. Dissemination of collection wide information in a distributed information retrieval system. In *SIGIR '95 Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 12-20, 1995.

[16] Voorhees, E. M., Gupta, N. K., and Johnson-Laird, B. Learning Collection Fusion Strategies. In *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 172-179, 1995.

[17] Voorhees, E. M. and Tong, R. M. Multiple Search Engines in Database Merging. In *Digital Libraries 97, The 2nd ACM International Conference on Digital Libraries*, Philadelphia. pages 93-102, 1997.

[18] Xu, J. and Callan, J. P. Effective Retrieval with distributed collections. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112-120, 1998.

[19] Xu, J. and Croft, W. B. Cluster-based Language Models for Distributed Retrieval. In *Proceedings of SIGIR '99: 22nd International Conference on Research and Development in Information Retrieval*, pages 254-261, 1999.

[20] Yuwono, B. and Lee, D.L. Server Ranking for distributed test retrieval systems on the Internet. In R. Topor and K. Tanaka, editors, *Proceedings of the Fifth International Conference on Database System for Advanced Applications (DASFAA)*, pages 41-49, Melbourne, 1997.