Database Selection Using Actual Physical and Acquired Logical Collection Resources in a Massive Domain-specific Operational Environment

Jack G. Conrad **Xi S. Guo** **Peter Jackson** **Monem Meziou**

TLR Research & Development
Thomson Legal & Regulatory
St. Paul, Minnesota 55123 USA
{*Jack.Conrad, Peter.Jackson*}*@WestGroup.com*

Abstract

The continued growth of very large data environments such as Westlaw, Dialog, and the World Wide Web, increases the importance of effective and efficient database selection and searching. Recent research has focused on autonomous and automatic collection selection, searching, and results merging in distributed environments. These studies often rely on TREC data and queries for experimentation. We have extended this work to West's online production environment where thousands of legal, financial and news databases are accessed by up to a quarter-million professional users each day. Using the WIN natural language search engine, a cousin to UMass's INQUERY, along with a collection retrieval inference network (CORI) to provide database scoring, we examine the effect that a set of optimized parameters has on database selection performance. We also compare current language modeling techniques to this approach. Traditionally, West's information has been structured over 15,000 online databases, representing roughly 6 terabytes of textual data. Given the expense of running global searches in this environment, it is usually not practical to perform full document retrieval over the entire collection. It is therefore necessary to create a new infrastructure to support automatic database selection in the service of broader searching. In this research, we represent our operational environment in two distinct ways. First, we characterize the underlying physical databases that serve as a foundation for the entire Westlaw search system. Second, we create a rearchitected set of logical document collections that corresponds to classes of high level organizational concepts such as jurisdiction, practice area, and document-type. Keeping the end-user in mind, we focus on performance issues relating to optimal database selection, where domain experts have provided complete pre-hoc relevance judgments for collections characterized under each of our physical and logical database models.

Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002

1 Introduction

The proliferation of online textual information, manifested in data residing on both the World Wide Web and in commercial data environments, has placed emphasis on the growing importance of database selection techniques. Effective and efficient database selection techniques are increasingly critical today because, given a user's query, it is simply no longer practical to perform comprehensive full-text searches over all of the underlying data collections.[1] Moreover, it is a growing challenge to provide high precision search results given the vast scope of the Web or commercial databases. This situation introduces the need for reliable, high performance distributed searching.

Because of the work of Gravano, Callan, French, and others [16, 5, 11, 26], aspects of distributed search have been divided into four principle activities: (1) collection ranking; (2) collection selection; (3) searching the chosen collections; (4) merging the results into a uniform set. Their approaches to these issues have made considerable progress in terms of autonomous systems functioning without user interaction. These experiments rely largely on fully automated approaches that include database selection as well as document retrieval and merging.

[1] In this paper, we will use collection to refer to a database of textual documents.

As an alternative to the four activites described above, Fuhr describes a theoretic model that comprises the first two steps and avoids the heuristic criteria of the second step by employing a broker that estimates the retrieval cost of each database. The broker in turn determines whether and how many documents to retrieve from each database [14].

In the majority of our user sessions, legal researchers are searching for information from a known, familiar source. As the practice of law has evolved over recent years, however, researchers are increasingly turning to extra-legal sources to supplement their legal research. Information venders such as West Group and Lexis-Nexis have supplied this demand with more business, medical and scientific information. Yet as these information domains move away from the traditional domain of the legal researcher, information providers need to offer additional assistance in choosing the appropriate sources.

We have subsequently conducted experiments in an environment that replicates our actual distributed production resources where hundreds of thousands of users access over 15,000 document collections daily. We use a collection retrieval inference network (CORI)[5] run against a set of physical database representations. The bedrock of our system is the WIN search engine[2] [30, 32, 31], a close relative to the INQUERY engine developed at the Center for Intelligent Information Retrieval at the University of Massachusetts [1, 2]. The performance of our system has led us to conclude that there is a role for automatic collection selection in the act of simplifying a user's interaction with massive data environments. In certain data environments, users are still required to remember cryptic database identifiers or abbreviations in order to gain access to their desired sources. We demonstrate that in massive online environments like Thomson's, state-of-the-art database selection techniques can simplify required access mechanisms while still delivering high performance results.

Our work is distinct in several significant ways. Much of the research performed thus far has been academic in nature rather than of production-caliber. Some of it appears to be unrepresentative of real production environments because (a) it is often too general for specific problems; (b) it does not necessarily scale to accommodate real problems; and (c) it is often driven by ideas rather than needs. Regarding the problem at hand, much of the database selection work performed to date has relied on data sets that (i) are of identical or very similar size; (ii) are discrete, having no overlapping documents; (iii) represent sets of hundreds, rather than thousands or tens of thousands of databases; (iv) contain quantities of documents in the 10K range, rather than in the 100K or 1,000K range; and (v) are characterized by an environment in the 100 GB range, rather than the terabyte range.

By contrast, our work distinguishes itself as:

- it represents tens of thousands of collections;
- its collections contain documents in the millions;
- a single document can appear in ten or more collections;
- its collection sizes vary by several magnitudes;
- its collections are designed to serve a real rather than generic domain;
- it represents an operational environment with data in the terabytes.

In mid-2000, analysis determined that there were in excess of two billion unique, publicly accessible "pages" on the Web, with an average of between 10-15KB per page [24, 21]. With a rate of growth of over seven million new pages added per day, the Web was on track to double by mid-2001 [24]. These figures indicate that there are currently in the range of 40 to 60 terabytes of indexable text on the Web. Since West Group's alliance with Dialog, their combined repositories now encompass roughly half that amount of data, corresponding to tens of thousands of databases. The majority of their new databases come from news and non-legal domains, in contrast with West's historically legal focus. Although computational resources permit comprehensive searches against global indexes–thus allowing users to be the final filter–the scope of the problem exacts a non-trivial cost. Recent experiments have focused on hundreds of collections, yet production environments provide over ten thousand databases, at times with an order of a million documents in each.

Given the challenges of our operational environment, our research objectives focus on the following.

1. Discover how existing approaches would perform in an industrial environment;
2. Examine competing database scoring models to determine while are most reliable, e.g.,
   (a) Collection Retrieval Inference Network vs.
   (b) Language Modeling.
3. Compare performance using an existing physical organization of databases with a rearchitected logical organization of databases.

The remainder of this paper is organized as follows: Section 2 reviews related work in collection selection and contrasts our work with its core focus. Section 3 describes our experiments and how we evaluated our methods. Section 4 addresses our database ranking algorithms and how these are distinguished from related

[2]WIN stands for *Westlaw Is Natural*.

approaches. Section 5 summarizes our results. We discuss special challenges associated with the Database Selection problem in Section 6 and present our conclusions in Section 7. In Section 8, we disclose plans for future research. Lastly, in Section 9, we express our gratitude to those contributing to this work.

2 Previous Work

The key components of distributed search have been divided into four activities: (1) collection ranking; (2) collection selection; (3) searching the chosen collections; (4) merging the results into a uniform set. In our experiments, like those of Hawking, Yuwono, and others, we investigate the first two activities [19, 38] and use the third as a validation step. Others such as Voorhees and Craswell, et al. have focused specifically on merging results [34, 9].

Gravano, et al. has called the challenge the *text database resource discovery problem* [17, 23]. By contrast, Callan, et al. has referred to this as the *collection selection problem* [5, 36]. Hawking, et al. has referred to this as a server selection issue [19, 38]. And French, et al. has termed this *database selection* [11, 26]. For clarity and generality, we follow the *database selection* terminology used in this last work and focus on all but the results merging activity described above, since there is evidence that improved collection recall can lead to improved overall distributed retrieval [36, 26].

Gravano, et al. was one of the seminal investigators of the database selection problem for large data environments, including the Internet, by way of a Glossary of Servers Server (GlOSS) [17, 16, 18]. Originally developed around a Boolean query retrieval model (bGlOSS), he subsequently generalized his approach to the vector-space retrieval model (vGlOSS) [16] and demonstrated that this framework and its associated *goodness* metric delivers effective document retrieval in large data environments, including the Internet [18]. In a related study, Callan, Lu and Croft addressed the difficulties of distributed vs. centralized information retrieval by harnessing UMass' INQUERY search engine. They applied its inference net retrieval model (CORI)[3] to multiple collections [5]. Other researchers expanded this investigation while using the INQUERY engine and CORI nets. Xu and Callan compared distributed with centralized retrieval and the role query expansion can play in effective distributed IR [36]. Powell, et al. further quantified distributed vs. centralized retrieval by comparing the performance of CORI against the best-case relevance-ranked retrieval (RBR) [26]. French, Callan, et al. also evaluated the scoring behind these models by comparing GlOSS's Goodness estimator and its so-called Ideal ranks with RBR [13] and later performed direct CORI vs. GlOSS and CVV[4] comparisons [11, 6]. CORI was the most accurate and stable of the three algorithms and had fewer problems with normalization biases than the other two [e.g., no. of docs (GlOSS) and doc. length (CVV)].

Many of these techniques require extensive knowledge of the term and concept distributions in available collections either directly [26, 36, 5] or through preliminary query-based sampling [4]. Some of these techniques suggest that a reorganization of large amounts of data, either by the clustering of different database selection indexes or by topical organization, may improve overall retrieval performance [37, 22, 12]. In massive online data environments where the stream of incoming data or the requirements for updates can be daunting, such infrastructure issues can best be addressed in a logical, rather than physical, manner in order to be practical for operational environments.

We have observed that few of these investigations have focused on issues related to domain utility, especially for those domains that rely on closed vocabularies. One recent exception is the work on query augmentation by French, et al. that used a MeSH (medical) closed vocabulary tied to the OHSUMED collection's MEDLINE articles [12]. Our work similarly needs to handle features of the legal domain, including the ability to treat legal citations (e.g., *583 Cal.App. 437*), legal titles (e.g., *Brown v. Board of Education*), and legal phrases (e.g., "the statute of limitations" non-equivalence to "limitations of the statute").

Overall, the above approaches have been responsible for considerable performance gains for autonomous systems with no user interaction [5, 36, 37, 17, 16, 18, 13, 11, 26, 38, 19, 23, 15]. Other approaches have asked users to provide metadata concepts or applied thesauri with semantic links to a query, either before or after examining highly-ranked source documents [7, 10, 20]. The overwhelming focus of these studies has been on completely automatic techniques, yet operational environments often also need to be able to accommodate "users in the loop." This is one of the overriding requirements for our system, which we return to in section 6.

3 Experimental Methodology

Our investigation consists of two phases. The first involves a series of database selection experiments performed on collections that correspond with physical data sets obtained directly from our production environment. The second consists of a different series of database selection tests, involving high-level logical data sets which were produced by consolidating preexisting lower-level production-side databases. The second phase is motivated by lessons learned in the first phase, focuses on databases designed largely along topical and jurisdictional lines, and is supplemented by subsequent *document* selection trials.

[3] CORI stands for "Collection Retrieval Inference Net."
[4] CVV is the abbreviation for Cue Validity Variance.

3.1 Data

3.1.1 Physical Databases

From our production environment we obtained 100 of approximately 1000 physical databases and their term frequency distributions. These consisted of roughly 100,000 (unstemmed) terms and 6,000 (stemmed) terms. In addition, we obtained the frequency distributions for roughly 500 legal phrases. The 100 physical databases we chose represented a broad cross-section of the 15,000 logical databases available to Westlaw's online clients. They include primary law (case law, statutes, and treatises), secondary law (law journals, reviews, and annotated compendiums), specialized resources (rulings and settlements, insurance and taxation documents, etc.), and news publications. In all, the 100 physical databases covered nearly 38% of the textual information available on Westlaw. (See Table 1) The creation of our physical databases is similar to data partitioning approaches invoked by other researchers [22, 37].

3.1.2 Logical Databases

In the second phase of our investigation, we constructed 128 high-level logical databases which cover virtually all of the important content available on Westlaw.[5] We thus reduced the complexity of Westlaw collections by a factor of two magnitudes (down from 15,000 databases). These collections were designed to address three metadata "views": (1) jurisdictional (to cover 60 state & federal-level court authorities); (2) practice area (e.g., bankruptcy, employment, international law, etc.); and (3) document-type (e.g., case law, statutes, law reviews, news, etc.). As such, they represent collections generated by a type of topical clustering and together comprise over 2 TB of data.

Unlike phase I, in phase II we did not have the benefit of pre-constructed production-side indexes. We consequently profiled these logical collections in two ways. First, we used *random sampling* of the collections, using 500, 1,000, and 2,000 documents. Secondly, we compared these results with those produced by a version of *query-based sampling* (QBS) [4] in which we submitted to each of the collections under consideration a single, short topic-specific, collection-relevant query to obtain our document set.[6]

Callan and Connell used the Spearman Rank Correlation Coefficient to determine the degree of similarity in the ranks of terms when comparing complete resource descriptors to acquired or learned resource descriptors [27]. He found that after 250 documents, about 80% of the word occurrences in the collection have already been found and that the new vocabulary being discovered is relatively rare [4]. Given larger databases, our smallest samples consisted of 500 documents, and at this level nearly all of our spearman coefficients tended to be at 0.8 or above when we could perform comparisons between physical and logical database correspondences (e.g., CTA, the "U.S. Court of Appeals, " "AZ-CA, Arizona Cases," etc).

Collection Information	Physical DBs	Logical DBs
No. of Collections	1000	128
Collections Profiled	100	128
Documents / Profile	All	500/1000
Avg. Docs / Collection	298,935	378,468
Avg. Tokens / Profile	97,299	22,296/47,450

Table 1: Data Set Characteristics

3.2 Queries

3.2.1 Original Test Set

We obtained five sets of 50 real user queries from our production-side database selection query logs. Our decision to rely on sets of 50 or more queries was guided by evidence that "for most [precision] measures, 50 [query] topics is sufficient to give an error rate less that 2% or 3% in these [IR] experiments" [3].[7] From these we also produced two additional sets of phrasal queries (20 queries in one; 10 in the other). These two query subsets permitted us to test our phrase recognition and processing techniques. Our queries were randomly selected with the minimum requirement that they be at least four terms in length.[8] The rationale for this minimum threshold is that if we could not handle queries of at least four terms, there is less likelihood that we could adequately handle shorter queries. In addition, we varied the minimum query length from one set of queries to another, as indicated in Figure 1 and Table 2. Average query length per query set was varied because we wanted to monitor to what extent performance would change relative to query length, since it has been shown that longer query statements reduce ambiguity associated with very short queries [28].

3.2.2 Logical Test Set

In the second phase, we wanted to strengthen our approach by using a query set that was characterized by (a) fewer positive relevance judgments per query (i.e., sparser hits) and (b) slightly longer queries (thus better simulating the issues-driven information needs that representative legal practitioners tend to submit). The resultant set of 100 queries was again derived from actual user query logs and has a slightly larger average query length than the combined set from the first phase (about 9 vs. 8 [Median: 8.0 vs. 7.0]; see Figure 2 and Table 2).

[5] Public Records were the only notable exception. This logical database content more than doubled the physical databases' coverage of Westlaw.

[6] A domain expert in a sponsoring dept. crafted these queries.

[7] *Topic* is used here in the sense of a Text REtrieval Conference (TREC) query subject [35].

[8] Stop words were included in the term count.

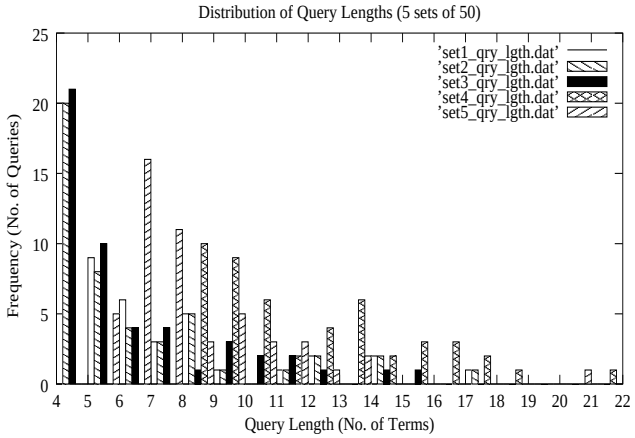

Figure 1. *Query Lengths (5 Sets) (Physical)*

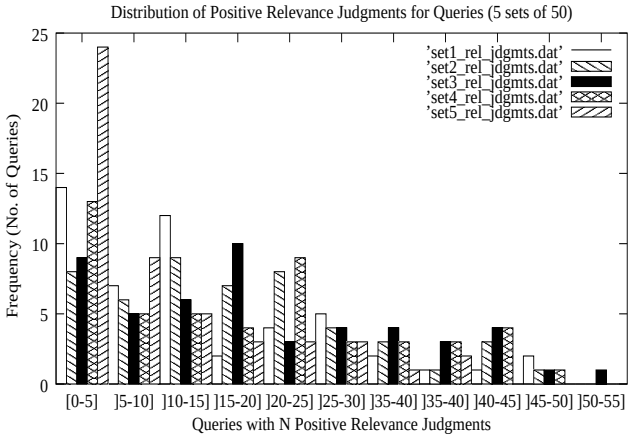

Figure 3. *Rel. Judgment Distrib. (5 Sets) (Physical)*

3.3 Relevance Judgments

Our query sets were judged by two attorneys with backgrounds in Library Science and who were well-acquainted with our data sets. Together they provided us with complete relevance judgments (5 sets × 50 queries × 100 physical collections or 25,000 judgments plus 1 set × 100 queries × 128 collections or 6,400 judgments). They made these judgments in a prospective (versus retrospective) manner.

Training sessions were held in order to ensure that the standard for judgments was consistent between the two domain experts from one data set to another. The first assessor largely trained the second assessor and reviewed her judgments. The first assessor provided judgments for the 5 sets of physical database queries. The second assessor provided judgments for the logical database queries. So queries for each phase had a single judge.

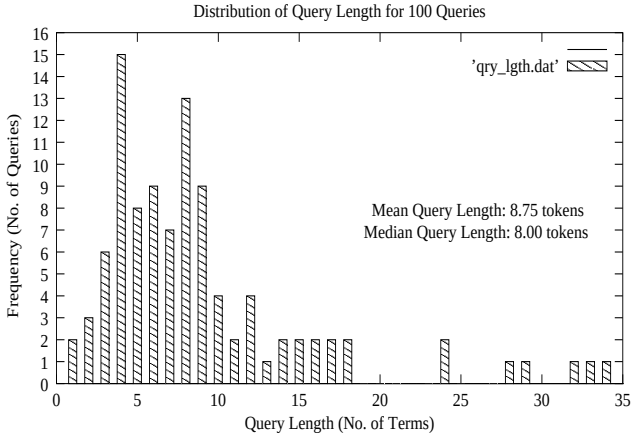

Figure 2. *Query Lengths (Logical)*

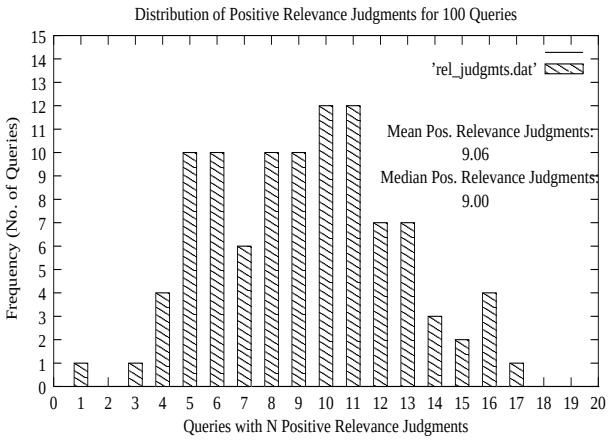

Figure 4. *Relevance Judgment Distrib. (Logical)*

Query Set	Number of Queries	Query Length		+Rel. Judgments/Qry		Top Average Precision
		Mean	Median	Mean	Median	
1.	50	6.2	5.0	15.1	12.0	40.3
2.	50	8.0	5.0	18.4	18.0	43.1
3.	50	6.1	5.0	20.3	19.0	46.6
4.	50	11.8	10.5	18.2	19.0	42.1
5.	50	7.8	7.0	10.1	6.0	40.6
Combined	250	8.0	7.0	17.0	14.0	42.3
Logical	100	8.8	8.0	9.1	9.0	53.9

Table 2: Query Set Characteristics

We proceeded with this approach to the judgments because there is evidence that "comparative evaluation of retrieval performance is stable despite substantial differences in relevance judgments" [33]. It should be noted that in these two phases, relevance judgments are made at the database-level only. Document-level relevance is examined in section 5.2.

Information on the distribution of relevance judgments for each of the query sets is included in Figures 3 and 4. As indicated in Table 2, the average number of databases judged relevant per query decreased for the logical database set (from 17.0 to 9.1) [Median: from 14.0 to 9.0].

For subsequent performance evaluation, we use average precision and precision at N profile documents (databases) for N up to 20.

4 Database Ranking

Much of the database selection research is based on applications of IR techniques to distributions of the terms and phrases that comprise the collections. Callan calls these distributions "complete resource descriptions" [4]. The metaphor invoked here suggests that each collection can be treated as a *meta-document* consisting of a set of collection-associated terms and phrases. The resulting database searched is thus the set of resultant meta-documents or collection profiles. It is assumed that the term statistics that characterize collections are readily available from the database indexes or can be approximated through the iterative use of probing queries [4]. It is also assumed to be too costly to query all the available databases, so either a fixed number of them are searched or a variable number whose score surpasses a preset threshold.

4.1 Collection Retrieval Inference Network

INQUERY's and WIN's algorithms for ranking documents have been previously reported [32], [2], [1]. In our case, the document retrieval model still holds, since we are working with meta-document representations of the collections in question. The $tf \cdot idf$ scoring model is applied to the database selection problem. Only now term frequency, tf, is replaced by document frequency, df, and inverse document frequency, idf, is replaced by inverse collection frequency, icf.

We have conducted experiments with three variations of *CORI net* scoring. Equation (1) presents the standard 'belief' score calculation for a Bayesian inference net retrieval engine like WIN, with its term frequency component, (tf), and inverse document frequency component, (idf). d_b is the minimum belief component. Equation (2) shows how the term frequency component is calculated, where d_t is the minimum term frequency component when term w_i is present in a collection representation, c_j. df_i is the number of documents the term w_i is found in and df_{max} is the number of documents containing the most frequent term in c_j. Equation (3) shows how the inverse document frequency component is determined, where cf represents the number of collection representations in which the query term w_i appears while $|C|$ is the total number of collection representations. We refer to these definitions as CORI Net1 scoring.

$$p_{belief}(w_i|c_j) = d_b + (1 - d_b) \cdot tf_b \cdot idf_b \quad (1)$$

$$where \quad tf_b = d_t + (1 - d_t) \cdot \frac{\log(df_i + 0.5)}{\log(df_{max} + 1.0)} \quad (2)$$

$$idf_b = \frac{\log(\frac{|C|+0.5}{cf})}{\log(|C| + 1.0)} \quad (3)$$

Equation (4) is a modified version of equation (2) where $df + K$ is substituted for df_{max}. It was inspired by experiments in document retrieval. Callan, Lu, and Croft defined K in terms of collection representation length together with parameters b and k, (5), where cw is the number of words in the collection and $\bar{cw}$ is the mean of cw in the collections being ranked [5].[9]

$$tf_b = d_t + (1 - d_t) \cdot \frac{df_i}{\log(df + K)} \quad (4)$$

$$K = k \cdot [\ (1 - b) + b \cdot \frac{cw}{\bar{cw}}\ ] \quad (5)$$

$$K = \alpha \cdot \frac{cw}{\bar{cw}} + \beta \quad (6)$$

In the rest of the paper, we refer to equation (5) as version CORI Net2 and our own equation (6) as version CORI Net3.

4.2 Language Modeling Approaches

In addition to our investigations involving Inference Net scoring, we conducted a series of parallel experiments using language modeling techniques. These were patterned after recent similar retrieval efforts [25, 29, 37]. We used the weighted sum approach (an additive model) to combine our language models.

$$P_{sum}(w|d) = \lambda \cdot P_{doc}(w|d) + (1 - \lambda) \cdot P_{database}(w) \quad (7)$$

where λ is a weighting factor between 0 and 1. A language model based only on a profile document may face sparse data problems when the probability of a word, w, given a profile *doc*, d, is 0 (an unobserved event). As a result, it may be useful to extend the original document model with a database model. An additive model can help by leveraging extra evidence from the complete collection of profiles. By summing in the contribution of a word, w, at this *database* level, we can mitigate the uncertainty associated with sparse data in the non-additive model.

[9] Callan, Lu and Croft found that $b = 0.75$ and $k = 200$ generally produced optimal results; by contrast, we found that $b = 0.6$ and $k = 300$ produced best results.

In addition, by treating the query as a sequence of terms, with each term viewed as an independent event and with the query representing the joined event, we have,

$$P_{sequence}(Q|d) = \prod_{m}^{i=1} P(w_i|d) \quad (8)$$

where $w_1, w_2, ...w_m$ is the sequence of terms in query Q. By treating the query as a sequence of terms in this way, as did Song and Croft [29], we are able to handle duplicate terms. Such treatment also permits the construction of a model with phrases in local contexts. Smoothing techniques were also incorporated to handle the issue of sparse data assocated with unobserved events [8], yet our results with smoothing surpassed those obtained from our WIN natural language engine only in phase I for physical databases.

5 Results

5.1 Database Selection Performance

We conducted two test phases on significantly different database infrastructures. The phases differ in terms of physical vs. logical organization, content-type vs. topic-type, and overall granularity. The first phase was performed upon 100 physical databases whose profiles were constructed directly from Westlaw collection indexes. Against these profiles, we tested our various scoring techniques. This approach helped us identify the best candidate scoring algorithms, albeit in a particular context. In the second phase, we focused on the most effective version of Cori Net scoring, comparing stemmed and non-stemmed data obtained through both random selection and query-based sampling. Results for the two phases are discussed below.

5.1.1 Physical Databases

In our experiments involving the physical databases, histograms (or "complete resource descriptions") of term frequencies were constructed from complete collection indexes. Performance was then measured based on optimizations made on an assortment of key variables.[10] Several scoring methods were also evaluated for their effectiveness, including variations of CORI and Language Modeling. Some of these results are shown on the next page. Performance fluctuations are shown for the five query sets consisting of 50 queries each.

Precision at the first recall point appears to vary between 60% to 80% (Figure 5). Upon closer inspection of precision at Nth database retrieved (Figure 6), performance at the top ranks is in the 40%-60%, which we would consider only marginally acceptable from a user perspective. These shallower gradients seem to underscore the fact that our physical databases are not organized along topical lines. There is thus a higher likelihood that relevant documents may be distributed among many collections.

We also examined the impact of phrasal processing in our physical database environment. Figure 7 shows six different phrasal combinations that were studied.[11] Although use of phrase recognition consistently produces the best results, performance improvements are generally most pronounced in the middle to lower recall points and often not statistically significant. So it is an open question how beneficial such added expense would be for an operational system.

5.1.2 Logical Databases

For our experiments involving logical databases, histograms (or "acquired resource descriptions")[12] of term frequencies were constructed from documents sampled in the actual collections. Callan concluded that "the resource descriptions created by *query-based sampling* are sufficiently similar to resource descriptions created from complete information that it makes little difference which is used for database selection," and, further, that "experimental results ... [are] robust with respect to variations in parameter settings" [4]. Given these findings, it is possible to view random sampling from cooperative systems as little more than a more progressive variant of query-based sampling. We examined both approaches and found that random document selection generally performed slightly better than QBS selection.

Because we were required to stem the acquired documents ourselves, rather than rely on production-provided stems, we needed to determine whether stemming would again outperform unstemmed term representations. Figure 9 illustrates that stemming continues to produce superior results to unstemmed collections, particularly at the top recall points. This was found to be true for virtually all pair-wise comparisons we ran. It was also determined that Cori Net3 using values of $\alpha = 1.0$ and $\beta = 400$ provided best overall performance (Figures 10 & 11). The baselines shown represent results from ranking databases strictly by size (no. of docs). Performance using the logical database infrastructure, especially precision at the top recall points, appears to surpass results produced by the physical databases (by as much as 10%, as indicated by Cori Net2 performance in Figures 8 vs. 12). We return to this comparison in Section 7. A performance enhancing post-retrieval process based on lexical patterns identified in the query is described in the next section.

[10] If not otherwise stated, results presented use stemmed collections, a term scaling factor of 20, a minimum term frequency threshold of 5, a stop word list of nearly 400 words, and cardinal numbers-only indexing (e.g., for publication years, etc). Each of these parameters was determined empirically.

[11] Because of the associated initial overhead, the phrasal tests focus on an important subset of the overall physical databases, namely, those containing judicial opinions (of which, there were 41 associated databases in all).

[12] Callan calls these "learned resource descriptions" [4]. We avoid this name since we did not use hundreds of QBS queries to obtain our sample, but rather, a short domain-relevant query.

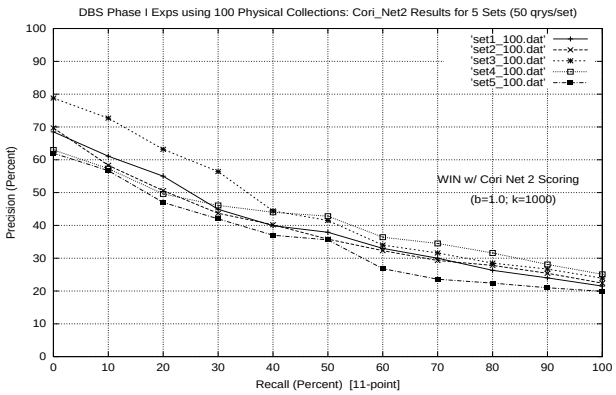

Figure 5. *Cori_Net2 Perf: Precision-Recall*

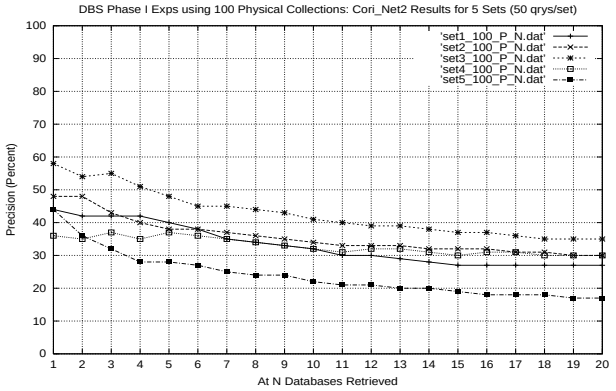

Figure 6. *Cori_Net2 Perf: Precision at N DBs*

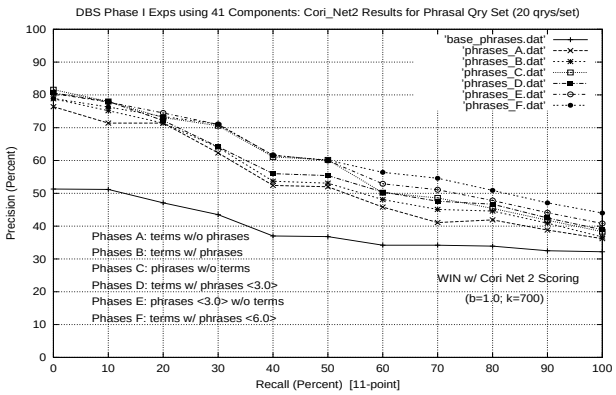

Figure 7. *Phrasal Performance*

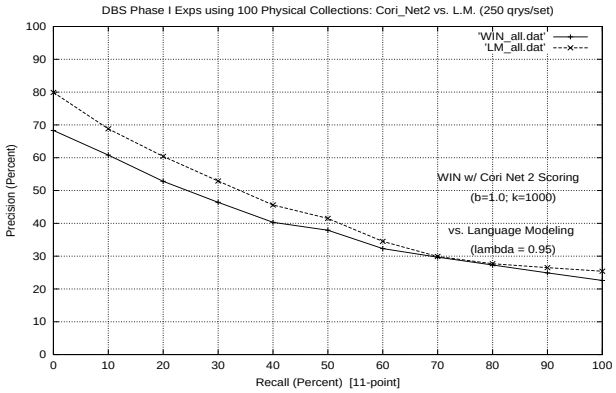

Figure 8. *Cori_Net2 vs. Lang. Model. Perf.*

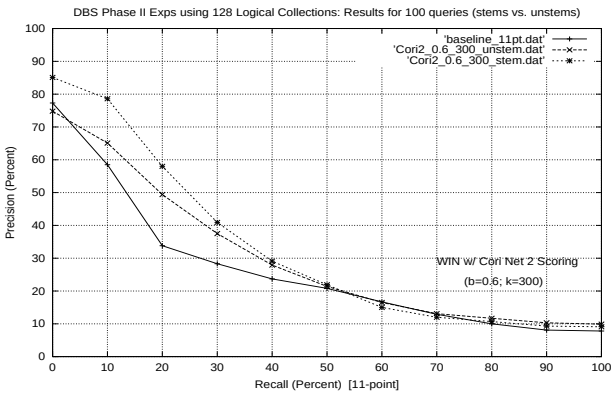

Figure 9. *Stemming vs. No Stemming Performance*

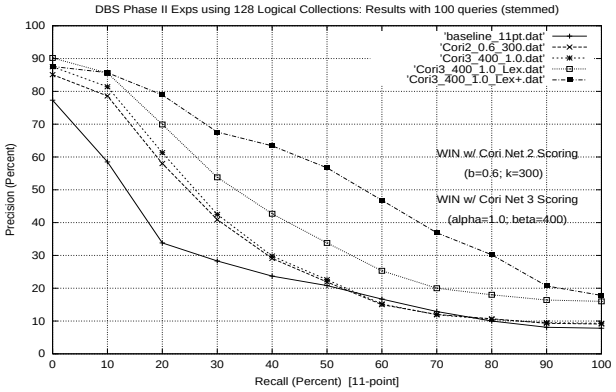

Figure 10. *Enh. Cori_Net3, Precision-Recall*

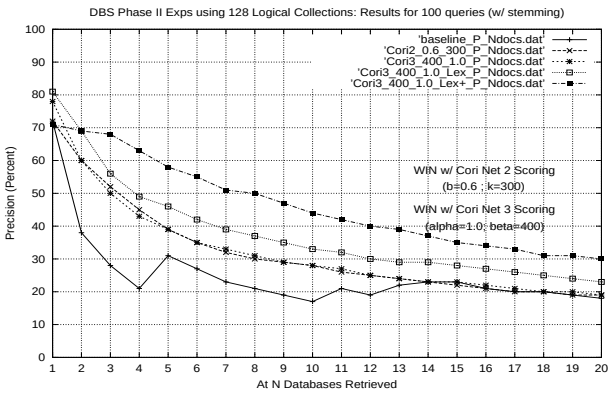

Figure 11. *Enh. Cori_Net3, Precision at N DBs*

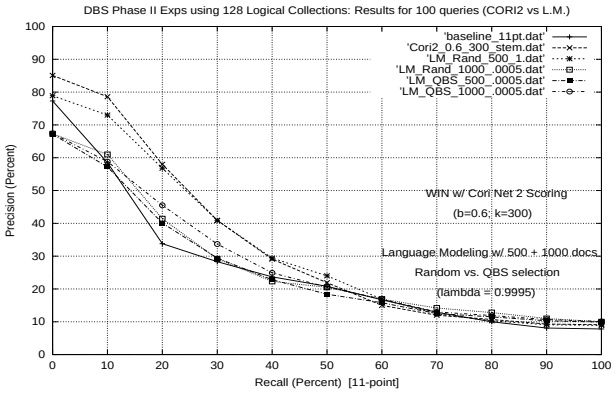

Figure 12. *Cori_Net2 vs. Lang. Model. Perf.*

5.1.3 Jurisdictional Lexical Analysis

Following the retrieval of a set of candidate databases, we discovered that performance can be enhanced by a post-process that lexically analyzes the query for *jurisdictionally relevant* content. That is, when no such context is found, then the results are reranked such that jurisdictionally biased collections are downweighted. The intuition behind this treatment is that legal *topics* are still going to appear in collections organized around, for instance, a state jurisdiction's corpus of judicial opinions (e.g., for a query like "Arizona Environmental Decisions."). This semantically intuitive process resulted in improvements in average precision of as much as 20% at top recall points (Figures 10 & 11). The plots marked 'Lex' apply the reranking only to those results for queries with no jurisdictional clues; those maked 'Lex+' apply the reranking to results for each query, but leave databases which match the lexical clues in their original rank(s).

Using the above results, we have also examined the performance of our database selection process at the query level, focusing on the role of query length. It is generally accepted that the shorter the query, the more difficult it is to disambiguate a query and produce high precision retrieval. In the legal domain, we generally believe that beyond a certain length, additional terms do not always ensure improved disambiguation. For instance, when queries change their focus from publications, regional courts, and practice areas to thorny legal issues, another level of difficulty can be introduced. Mean average precision in Table 3 tends to support such behavior. As length of the queries increases (following the removal of stopwords), so does the average precision, up to roughly a dozen terms, after which performance drops. One sees a similar trend with mean top-10 database precision, although performance tends to drop earlier. These results tend to correspond to our expectations, though given the small sizes of the last three categories, no definitive conclusions should be drawn from these samples.

5.1.4 Language Modeling Approaches

We discovered that our language modeling approach worked best with our stemmed *physical collections*, slightly surpassing the performance of WIN. When we compare CORI Net2 scoring, implemented by our WIN engine, to a Language Modeling approach using the combined set of 250 queries (Figure 8), we see that L.M. clearly outperforms WIN by at least 10% at the top recall points (although the overall performance difference is not statistically significant at the 0.05 level using the Wilcoxon test).

One of the reasons for this result is that the amount of data used to construct the model was substantial, representing extremely large collections and associated documents, with almost no overlap of content between these collections. The vast majority of query terms could be found in the model; hence, sparse data was minimized and little smoothing was necessary.

By contrast, the performance of our language model for the profiled *logical collections* degraded relative to WIN (Figure 12). In this particular case (*query-based sampling* vs. *random* doc selection using 500 and 1000 documents), Random_500 outperforms the other L.M. representatives, while QBS_1000 managed to outperform Random_500 for the middle recall points, but the overall differences are not statistically significant. In general, we witnessed little evidence that QBS selection warranted the added effort relative to Random selection. As the above results suggest, we found evidence that QBS may continue to improve slightly with additional documents while performance from Random selection appears to stabilize sooner.

One of the reasons for this degradation in L.M. performance relative to WIN is the increase in sparse data resulting from the reduced number of documents used by the logical database profiles [*O(1000) vs. O(100,000)*]. As a result, smoothing is invoked more often in phase II and allocates far more of the probability mass to unobserved events. The smoothing variation that we used consisted of a default probability generated from an attenuated corpus-wide average term frequency (to avoid potential problems with high default probabilities [29]). Another reason for this decreased performance is that there were no requirements for data isolation among the logical collections. So it is possible that a document represented under, for example, the state of *New Hampshire* might also appear under one of the topical collections, for instance, *bankruptcy*. Although there is little likelihood of this occurring in our profiles, the point is that the language representative of a legal practice area like *criminal law* may very well appear in other representations such as at the jurisdictional level.

WIN was more robust for the logical collections and generally outperformed our language model on these collections. As a result, we opted to use WIN in our final model for two reasons. The first is that it outperformed the L.M. in most instances. The second reason is that WIN was already available in our operational environment and did not require extensive reengineering, validation, and QA work to be brought online.

Query Lgth [Range]	No. of Qrys	Mean Avg Prec	Mean Top-10 Prec	Mean + Rel Jdgmt
[1- 3]	25	48.5	36.0	9.24
[4- 6]	50	49.8	39.0	9.26
[7- 9]	16	53.2	44.4	9.81
[10-12]	3	55.4	36.7	7.00
[13-15]	3	50.2	33.3	7.67
[16-]	3	52.9	46.7	9.00

Table 3: Query Length Performance Characteristics

5.2 Document-level Performance

To obtain a sense of the quality of the associated documents retrieved from our top database selection methods, we inspected several document sets retrieved from the top collections. We performed an evaluation of *document-level* precision involving 25% of our final query set for logical databases. The results of this evaluation are preliminary, yet suggestive of the performance users might expect. There were 100 queries in our final set for the logical databases. We selected 25 of them randomly. The top 20 documents returned from each of the 5 top-ranked collections were examined for relevance by a paralegal in our lab.

In all, 2,500 documents were reviewed (25 queries $\times$ 5 databases/query $\times$ 20 documents/database). To help clarify matters of degree, multi-valued relevance judgments were used. The four distinct judgments were: *on point* (i.e., of highest relevance), *relevant*, *marginally relevant*, and *non-relevant*. Multi-valued relevance judgments were used for at least two reasons. One was to make it easier on the paralegal/assessor to mitigate borderline judgments. The second was to obtain a finer-grained notion of the document-level precision. The results obtained are shown in Table 4.

Over half of the documents judged for relevance were found to be on point and fully three-quarters of the documents reviewed were found to be relevant or better. When one loosened the notion of relevance to include marginally relevant, totals exceeded 80%. The suggestion is that even when a fixed database cutoff is used for document retrieval, the results can be satisfactory from a user perspective. We would expect that as more sophisticated means of database cutoffs are implemented, such as inter-database scoring gap thresholds, performance could again improve significantly.

6 Discussion

The approaches we have examined for our physical and re-architected logical collections represent two alternative techniques for effective and efficient database selection in a distributed environment. One of our objectives was to test the viability of a re-architected infrastructure for the information system in order to ensure higher performance database selection.

In several different ways, the re-architecting of Westlaw content offers the information system a streamlined and effective means of handling user queries. In the past, users were expected to be able to refer to one or more unique database identifiers from a pool of thousands.[13] At one time, quantity of databases was considered only an asset. Tools and systems were subsequently designed to assist users to determine which identifiers corresponded to their relevant data sets. Granularity of the databases may have depended on how documents happened to be partitioned. Partitioning was sometimes based on hardware storage capabilities as much topical or jurisdictional organization. By rearchitecting the content with users of the system in mind, the number of databases required has been greatly reduced, and the organizational backbone is now based on a much more practical if not intuitive structure.

[13] Although such granularity can be a true asset to a specialist, it also has its distinct disadvantages, either for first-time users or for practitioners working outside their area of core competence.

6.1 Alternative Operational Systems

A high-level logical approach to database selection can be implemented in an online environment in at least two distinct ways. The first would be a completely autonomous approach in which a user's query would initially be used to determine the collection ranks based on probable relevance: the query would subsequently be run automatically against (a) the top n-ranked collections returned (where n was set to a fixed number like 5) or (b) the top collections where the "gap" between scores of two sequentially ranked collections remained lower than a certain fixed value. This would simulate the kinds of independent searches users are increasingly accustomed to running on the World Wide Web. Such a fully autonomous approach may also be suitable for younger, less experienced practitioners who have not yet developed abilities in crafting specialized Boolean or elaborate natural language queries.

Alternatively, once the top-ranked databases are determined, the system could present them to the user, again ranked by probable relevance, and ask the user to select the most relevant or most promising collections. Once the user selects one or more databases, the system could then perform a multi-database search on those collections. Each of these approaches has advantages and disadvantages. The advantage of the first approach is complete automation and speed as no intermediate steps are required. The disadvantage associated with this approach is, of course, that the user has no opportunity to deselect candidate databases that may clearly be off-topic and non-relevant to the user. The second approach has the potential to eliminate this disadvantage, but at the expense of time and user involvement. Furthermore, if the user is uncertain about the contents of any of the databases (e.g., if not enough information is available about the given collections), then the user might be better off relying on the general automatic system.

7 Conclusions

Through an extensive series of experiments conducted using representations of both physical and restructured logical databases, we have determined that either approach may achieve reasonably fair collection selection results in an operational environment consisting of many thousands of databases. We have corroborated certain findings of related research that often uses simulated databases consisting of segmented

No.	Relevance Type	Quantity	Percentage	Percentage Relevant or Marginally Relevant (Cum.)
1.	On Point	1415	56.60%	56.60%
2.	Relevant	439	17.56%	74.16%
3.	Marginally Relevant	199	7.96%	82.12%
4.	Not Relevant	447	17.88%	—
Total	Combined	2500	100.00%	82.12%

Table 4: Document-level Relevance Assessments

TREC data [35].

Unlike previously reported work, our research distinguishes itself by (i) representing data sets of dissimilar size (among both our physical and logical collections); (ii) permitting documents to appear in more than one collection (occurs less than 2% of the time in our two phases); (iii) representing on the order of 15,000 virtual databases that occur in an existing operational environment; (iv) marshaling databases containing hundreds of thousands rather than tens of thousands of documents; and (v) cumulatively representing data in the terabyte rather than gigabyte range.

We have asserted, however, that our most effective results are produced when using our logical databases, despite the fact that they possess roughly half as many positive relevance judgments as the physical databases. One probable reason for this performance is that, unlike with the physical collections, logical databases include differentiation along *topical* lines. Because of differences in (data) architecture and granularity and (query) average length and number of relevance judgments, it is not possible to perform a direct comparison between our physical and logical databases; nonetheless, by comparing performance results in Figures such as 6 vs. 11 (Precision at N Databases), we conclude that the new infrastructure and the logical collections generally deliver higher precision results. That these databases were developed with the end users' needs and familiarities in mind is an added benefit.

In terms of the actual scoring methods, we observed that over a wide-range of profiled data collections, simpler scoring algorithms like CORI Net3 performed, overall, better than those with layered tuning parameters like CORI Net2. Furthermore, the inclusion of lexical analysis (e.g., for jurisdictional clues) in query processing promises to improve baseline CORI Net performance significantly. In addition, our experiments that investigated the comparative performance of language modeling indicated that language modeling approximated or surpassed the performance achieved by our natural language engine (WIN) for our physical collections, yet resulted in an inferior performance when run against our logical collections. Our tests indicated that language modeling had to rely on smoothing much less frequently with our complete physical database resources than it did with our acquired (profiled) logical database resources. Consequently, its performance deteriorated relative to WIN for our logical databases.

Such front-end database selection processing can contribute significantly to the efficiency of large online systems with hundreds of thousands of users and tens of thousands of traditional data sources. Our longer term view is to integrate such approaches into a suite of collection selection tools, both conventional and domain-driven. It would ultimately be up to users to determine which approach would be most appropriate for a given information need. Over time and with experience, they will best be able to judge, based on the granularity and context of the query, what would be the most reasonable technique or tool to invoke.

8 Future Work

There has recently been an increased interest in clustering as a means of improving precision for users [39]. Given that our rearchitected logical databases represent document sets categorized at a high level, clustering techniques may offer another mechanism of producing useful sub-categories under some of our more general logical databases. This is what we plan to explore in our next phase. We also intend to examine the effectiveness of more robust smoothing techniques to determine whether they may contribute to improved performance of our language models, especially when applied to acquired (profiled) logical database resources. We will expand our investigation of actual document-level relevance as well.

9 Acknowledgements

We thank Shakila Xavier for her role in our extensive test suites. We thank Jon Eveslage for developing the production processes to generate our database profiles. We are grateful to Joanne Claussen and Pauline Afuso for assessing collection-level relevance for these tests, and to Joanne again for her contributions to the design of our logical database infrastructure. And much thanks goes to Dan Dyke for his work in assessing document-level relevance.

10 References

[1] J. Allan, J. Callan, W. B. Croft, L. Ballesteros, D. Byrd, R. Swann, and J. Xu. INQUERY does battle with TREC-6. In *Proc. of the Sixth Text REtrieval Conf.*, pages 169–206. TREC, Nov. 1997.

[2] J. Broglio, J. Callan, and W. B. Croft. INQUERY system overview. In *Proc. of the TIPSTER Text Program (Phase I)*, pages 47–67. TIPSTER, 1993.

[3] C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *Proc. of SIGIR '00*, pages 33–40. ACM Press, July 2000.

[4] J. Callan and M. Connell. Query-based sampling of text databases. In *ACM Trans. on Information Systems (TOIS)*, pages 97–130. ACM Press, April 2001.

[5] J. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proc. of SIGIR '95*, pages 21–29. ACM Press, July 1995.

[6] J. Callan, A. L. Powell, J. C. French, and M. Connell. The effects of query-based sampling on automatic database selection algorithms. Tech Report CMU-LTI-00-162, Language Technologies Institute, School of C. S., Carnegie Mellon Univ., 2000.

[7] A. S. Chakravarthy and K. B. Haase. Netserf: Using semantic knowledge to find Internet information archives. In *Proc. of SIGIR '95*, pages 4–11. ACM Press, July 1995.

[8] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proc. of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, pages 310–318. Morgan Kaufmann, June 1996.

[9] N. Craswell, D. Hawking, and P. Thistlewaite. Merging results from isolated search engines. In *Proc. of the 10th Australasian DB Conf. (ADC-99)*, pages 189–200. Springer-Verlag, Jan. 1999.

[10] R. Dolan, D. Agrawal, L. Dillon, and A. E. Abbadi. Pharos: A scalable distributed architecture for locating heterogeneous information sources. Tech Report TRCS95-05, Univ. of California–Santa Barbara, Dept. of C. S., July 1996.

[11] 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. of SIGIR '99*, pages 238–245. ACM Press, Aug. 1999.

[12] J. C. French, A. L. Powell, F. Gey, and N. Perelman. Exploiting a controlled vocabulary to improve collection selection and retrieval effectiveness. In *Proc. of CIKM '01*, pages 199–206. ACM Press, Nov. 2001.

[13] J. C. French, A. L. Powell, C. L. Viles, T. Emmitt, and K. J. Prey. Evaluating database selection techniques: A testbed and an experiment. In *Proc. of SIGIR '98*, pages 121–129. ACM Press, Aug. 1998.

[14] N. Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems (TOIS)*, 17(3):229–249, July 1999.

[15] R. Goldman, N. Shivakumar, S. Venkatasubramanian, and H. García-Molina. Proximity search in databases. In *Proc. of VLDB '98*, pages 26–37. Morgan Kaufmann, Aug. 1998.

[16] L. Gravano and H. García-Molina. Generalizing GlOSS to vector-space databases and broker hierarchies. In *Proc. of VLDB '95*, pages 78–89. Morgan Kaufmann, Sept. 1995.

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

[18] L. Gravano, H. García-Molina, and A. Tomasic. GlOSS: Text-source discovery over the Internet. *ACM Transactions on Database Systems (TODS)*, 24(2):78–89, June 1999.

[19] D. Hawking and P. Thistlewaite. Methods for information server selection. *ACM Transactions on Information Systems (TOIS)*, 17(1):40–76, Jan. 1999.

[20] M. A. Hearst. Using categories to provide context for full-text retrieval results. In *Proc. of the RIAO (Computer Assisted Information Retrieval) Conference*, pages 115–130. RIAO, Oct. 1994.

[21] A. Heydon and M. Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, Dec. 1999.

[22] L. Larkey, M. Connell, and J. Callan. Collection selection and results merging with topically organized U.S. patents and TREC data. In *Proc. of CIKM '00*, pages 282–289. ACM Press, Nov. 2000.

[23] W. Meng, K.-L. Liu, C. Yu, Wang, Y. Chang, and N. Rishe. Determining text databases to search in the Internet. In *Proc. of VLDB '98*, pages 14–25. Morgan Kaufmann, Aug. 1998.

[24] B. H. Murray and A. Moore. Sizing the Internet. A white paper, Cyveillance, July 2000.

[25] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proc. of SIGIR '98*, pages 275–281. ACM Press, Aug. 1998.

[26] A. L. Powell, J. C. French, J. Callan, M. Connell, and C. L. Viles. The impact of database selection on distributed searching. In *Proc. of SIGIR '00*, pages 232–239. ACM Press, July 2000.

[27] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, 2nd edition, 1992.

[28] M. Sanderson. Word sense disambiguation and information retrieval. In *Proc. of SIGIR '94*, pages 142–151. Springer-Verlag, July 1994.

[29] F. Song and W. B. Croft. A general language model for information retrieval. In *Proc. of CIKM '99*, pages 316–321. ACM Press, Nov. 1999.

[30] P. Thompson, H. Turtle, B. Yang, and J. Flood. TREC-3 ad hoc retrieval and routing experiments using the WIN system. In *Proc. of the Third Text REtrieval Conference (TREC-3)*, pages 211–217. NIST, Nov. 1995.

[31] H. Turtle. Natural language vs. Boolean query evaluation: A comparison of retrieval performance. In *Proc. of SIGIR '94*, pages 212–221. Springer-Verlag, July 1994.

[32] H. R. Turtle. *Inference Networks for Document Retrieval*. Ph.d. diss., Univ. of Mass.–Amherst, 1991.

[33] E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proc. of SIGIR '98*, pages 315–323. ACM, Aug. 1998.

[34] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning collection fusion strategies. In *Proc. of SIGIR '95*, pages 172–179. ACM Press, July 1995.

[35] E. M. Voorhees and D. Harman. Overview of the Sixth Text REtrieval Conference (TREC-6). *Information Processing and Management*, 36(1):3–35, Jan. 2000.

[36] J. Xu and J. Callan. Effective retrieval with distributed collections. In *Proc. of SIGIR '98*, pages 112–120. ACM Press, Aug. 1998.

[37] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *Proc. of SIGIR '99*, pages 254–261. ACM Press, Aug. 1999.

[38] B. Yuwono and D. L. Lee. Server ranking for distributed text retrieval systems on the Internet. In *Proc. of the 5th Annual Int'l Conf. on Database Systems for Advanced Apps. (DSAA-97)*, pages 41–49. World Scientific Press, April 1997.

[39] Y. Zhao and G. Karypis. Improving precategorized collection retrieval by using supervised term weighting schemes. Tech Report TRCS01-43, Univ. of Minn., Dept. of C. S., Oct. 2001.