Methods for Information Server Selection

DAVID HAWKING and PAUL THISTLEWAITE
Australian National University

The problem of using a broker to select a subset of available information servers in order to achieve a good trade-off between document retrieval effectiveness and cost is addressed. Server selection methods which are capable of operating in the absence of global information, and where servers have no knowledge of brokers, are investigated. A novel method using Lightweight Probe queries (LWP method) is compared with several methods based on data from past query processing, while Random and Optimal server rankings serve as controls. Methods are evaluated, using TREC data and relevance judgments, by computing ratios, both empirical and ideal, of recall and early precision for the subset versus the complete set of available servers. Estimates are also made of the best-possible performance of each of the methods. LWP and Topic Similarity methods achieved best results, each being capable of retrieving about 60% of the relevant documents for only one-third of the cost of querying all servers. Subject to the applicable cost model, the LWP method is likely to be preferred because it is suited to dynamic environments. The good results obtained with a simple automatic LWP implementation were replicated using different data and a larger set of query topics.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*distributed databases*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process*; *selection process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*information networks*; H.3.6 [**Information Storage and Retrieval**]: Library Automation—*large text archives*

General Terms: Design, Experimentation, Performance

Additional Key Words and Phrases: Information servers, Lightweight Probe queries, network servers, server ranking, server selection, text retrieval

1. INTRODUCTION

The problem of locating relevant text documents from distributed network servers is partially solved by large-scale centralized indexing services such as Alta Vista and Excite. This centralized model suffers from four limitations. First, even systems with enormous capacity are likely to index only a

The authors wish to acknowledge that this work was carried out within the Cooperative Research Centre for Advanced Computational Systems established under the Australian Government's Cooperative Research Centres Program.

Authors' address: Cooperative Research Centre for Advanced Computational Systems, Department of Computer Science, Australian National University, Canberra, ACT 0200, Australia; email: dave@cs.anu.edu.au; pbt@cs.anu.edu.au.

fraction of all documents on the Internet. Examples of documents not indexed include those barred by robot exclusion, those missed by the indexing robot, those served by parameterized scripts, and those accessible only through a search service operated by the server on which they are located. Second, index information may be out-of-date. Third, a serious researcher has no effective means of restricting searches to authoritative primary sources. Finally, public Internet indexing services do not index documents on private "intranets."

Various studies (summarized below) have examined variants of an alternative distributed model in which a user's search requests are forwarded by a *broker* to a carefully selected subset F of a long list S of known *servers*.

S is not likely to include all servers on the Internet but rather a shorter list prepared by the searcher's organization or by an Internet search service. S may include, or entirely consist of, servers on an organization's private network.

In the simplest distributed model, all servers operate a query-processing service restricted to the documents which they serve. This model is assumed throughout the present article.

The broker endeavors to merge the results obtained from the servers in F into a ranked list which will best meet the user's need. The main reasons for selecting a subset of S are reduction of network or server access costs and improvement in the timeliness of result delivery. However, it is possible that a highly effective server selection method might also result in a better combined ranking of documents. The relationship between search effectiveness (precision-recall performance) and cost of searching is of particular interest.

In general, successful implementation of the broker model requires solutions to the following problems:

(1) How to translate the user's statement of information need into the query languages of the respective search servers (Query Translation).

(2) How to select the members of subset F (Server Ranking/Server Selection).

(3) How to merge search results from the different servers so as to achieve precision-recall goals. (Result Merging, also known as the Collection Fusion problem).

The interesting problems of Query Translation and Result Merging are beyond the scope of the present work, where the focus is on Server Ranking and Selection. Empirical comparisons of server ranking methods reported here avoid the query translation problem by assuming that all servers operate the same retrieval system. Similarly, a special type of relevance scoring (described below) is used to obtain server ranking results which are not confounded by incorrect result merging.

1.1 Terminology

Elsewhere in the literature, the terms *server*, *source*, and *subcollection* have been used synonymously. Here, following (for example) Yuwono and Lee [1997], the term *server* is chosen for use in this context, and the term *source* is reserved for describing the organization which created or supplied a collection of documents, e.g., Associated Press. The term *subcollection* is not used as it does not convey the idea that the data may be distributed across a network.

The literature uses many different terms for the entity which refers a user query to a subset of available servers. Examples include *meta search engine*, *broker*, *metabroker*, *search manager*, *receptionist*, *metaservice*, and *query intermediary*. Following (for example) Gravano and García-Molina [1996], the term *broker* is somewhat arbitrarily chosen for use here.

Following the TREC convention [Voorhees and Harman 1996], the English language statement of a user's information need is called a *topic description*. Topic descriptions may directly serve as queries for a retrieval system or may be converted into a system-dependent query language.

The term *perfect merging* describes an unrealistic system capable of combining the document lists returned by multiple servers into a merged list with all relevant documents at the head. By contrast, the term *correct merging* refers to a merging process which is capable of producing a merged ranking effectively identical to that which would have been produced had all documents from all selected servers been searched as a single collection by a single retrieval system. Correct merging requires that document scores produced by independent servers are strictly comparable, which almost certainly requires that all servers use the same document-scoring algorithm.

1.2 Summary of Relevant Literature

Several approaches to Server Selection use term frequency data to rank servers. Callan et al. [1995] use Collection Retrieval Inference (CORI) Nets whose arcs are weighted by term document frequency (df) and inverse collection frequency (icf). Having ranked servers, they use a clustering method to determine how many servers to consult.

The gGlOSS system of Gravano and García-Molina [1996] allows for a hierarchical broker structure in which first-level brokers characterize servers according to term dfs and aggregated term weights and where second-level brokers characterize first-level brokers in the same fashion. Servers need to know how to contact all first-level brokers, and first-level brokers need to know how to contact all second-level brokers, in order to keep information up-to-date. The goal is to use the information held by the brokers to estimate the number of relevant documents likely to be held by each server. This is done according to two alternative extreme-case hypotheses about term cooccurrence, which are experimentally compared.

Yuwono and Lee [1997] describe a centralized broker architecture in which the broker maintains df tables for all servers. The variance of df

values across servers is used to select terms from the user query which best discriminate between servers, and then servers with higher df values for those terms are selected to process the query. Servers must transmit changes in df values when they occur and therefore need to know the address of the broker.

By contrast, a number of other approaches to the Server Selection problem are based on the use of server descriptions. Chakravarthy and Haase [1995] propose a method for locating the Internet server containing a *known item*. They manually characterize a large set of servers using semantic structures expressed in WordNet [Miller 1995] style. For each new search specification, a semantic distance based on WordNet hyponymy trees is computed for each server, and the server with the smallest such distance is chosen. They also use the SMART retrieval system to select servers based on unstructured, manually generated server descriptions and compare performance of the two methods. Kirk et al. [1995] propose the use of Knowledge Representation technology to represent both queries and server content and have developed algorithms for choosing a necessary and sufficient subset of servers described in this way.

The Pharos system proposed by Dolin et al. [1996] uses decentralized, hierarchical metadata descriptions for selecting appropriate servers. The authors suggest generating metadata automatically by performing cluster analysis of the server data and classifying the clusters within a manually defined metadata taxonomy.

Finally, isolated Database Merging approaches assume that neither global collection statistics nor server descriptions are available. The two approaches taken by Voorhees et al. [1995] rank servers using information derived from past query processing. In one, similarities are computed between the weighted term vector representing a new search topic and vectors representing each available past topic. The relevant-document distribution for the average of the k most similar past queries is used to score the utility of servers. In the other approach, vectors representing past queries are clustered into groups believed to represent topic areas, and similarities are computed between the new topic vector and the centroids of each past topic cluster. It should be noted that Voorhees and her colleagues do not attempt to restrict the number of servers accessed but rather to determine the optimum number λ_i of documents to retrieve from each server i, based on usefulness of servers to past research topics.

1.3 Overview of the Present Article

The present study compares server selection methods capable of operating in the absence of both global collection information and server descriptions. The methods studied do not require servers to have knowledge of brokers. Three methods based on historical data, including a variant of the Voorhees et al. [1995] methods, are empirically compared with a novel *Lightweight Probes* method (described below) that requires no past information. Comparisons are made using TREC [Voorhees and Harman 1996] data, research topics, and relevance judgments.

Section 2 details the experimental framework used in the present experiments. Section 3 contains an evaluation of each of the methods in turn. These evaluations are followed by comparisons of the estimated best-possible performance of each method (Section 3.6), comparisons of performance of practical implementations of each method (Section 3.7), and comparisons of automatic implementations (Section 3.8). Supplementary experiments using different data and a much larger set of queries are presented in Section 3.9 to confirm the generality of results obtained using automatically generated Lightweight Probes. Section 3.10 discusses cost models, compares server selection methods on the basis of equal estimated cost, and addresses cost-benefit issues. Section 3.11 briefly addresses the problem of choosing the number of servers to access. Finally, Section 4 discusses the performance and applicability of the methods studied and identifies a number of areas for further research.

2. EXPERIMENTAL FRAMEWORK

In essence, the evaluation methodology was as follows. A large test collection of text documents was distributed across approximately 100 simulated servers. The test documents had been previously judged for relevance to a set of topics. For each topic, a server ranking was generated using the method to be evaluated, and the cumulative numbers of relevant documents held by the servers up to given points in the ranking were calculated. Server selections were compared on the basis of relative proportion of relevant documents held, or alternatively, on the basis of the ratio of precision/recall performance of merged retrieval results from the selected subset to that achieved by merging results from all servers.

Four different basic approaches to server ranking were evaluated: Server General Utility (Section 3.2), Collection Promise (Section 3.3), Topic Similarity (Section 3.4), and Lightweight Probes (Section 3.5). Manual and automatic versions of the latter two were implemented and studied. These approaches were compared with two controls: Random, representing a performance floor, and Optimal (see Section 3.1.2), representing a performance ceiling. The availability of TREC-5 [Voorhees and Harman 1996] data and relevance judgments enabled accurate evaluation of the performance of each method over a large number of retrieval topics. Given a distribution of TREC documents across servers it is possible to know the locations of all the documents relevant to each topic.

Server rankings were evaluated in two ways. The first assumed an ideal retrieval system on each server and resulted in a theoretical measure of the effectiveness of the ranking. Unfortunately, the theoretical measure can be based only on the Recall (proportion of relevant documents retrieved) dimension and gives no information about Precision (proportion of retrieved documents which were relevant). Accordingly, server ranking methods were also compared empirically, using actual queries and a real retrieval system on each server. Definitions of the measures used are given in Section 2.6.

For each method and variant, actual retrieval runs were performed over four different server subset sizes, corresponding to one-half (49), one-third (33), one-fifth (20), and one-tenth (10) of the total number (98) of servers. Each retrieval run processed 50 queries and was evaluated in comparison to the performance achieved by the same queries processed over all servers.

Finally, it was recognized that server selection methods may be implemented in a range of different ways and that different implementations may vary in performance. Accordingly, lower-bound estimates of the best-possible performance achievable by any implementation of the methods are derived using “hindsight” and compared with actual manual and automatic implementations. For example, the best-possible performance of Topic Similarity is estimated by using a topic similarity measure based on the known distributions of relevant documents rather than on the text of the topic.

Server selection methods were compared on the basis of equal numbers of servers accessed. The problem of choosing an optimum number of servers for a given user and a given information need is considered to be independent of the server ranking method. (See Section 3.11.)

2.1 Text Data

The text data used in the experiment comprised TREC CDs 2 and 4, representing a total of approximately 2 gigabytes divided into 524,929 documents. Six distinct collections (each corresponding to a distinct document source) were represented: Associated Press (11), US Federal Register (28, comprising 1988 and 1994 groupings), Wall Street Journal (14), Ziff-Davis computer publications (8), Financial Times (26), and US Congressional Record (11). Each collection was divided across the number of servers shown in parentheses. The total amount of data (measured in bytes) on each server was roughly constant, although the number of documents per server varied from 1,548 to 8,527, with a mean value of 5,356. The division of data was as specified in the TREC-5 Database Merging task. The distribution of number of documents per server is shown in Figure 2. No server held data from more than one collection.

2.2 Hardware and Software

The experiments reported below were carried out using the PADRE [Hawking and Bailey 1997; Hawking and Thistlewaite 1995] text retrieval system running on a 128–node parallel distributed-memory machine, the Fujitsu AP1000. Details of this machine are given by Horie et al. [1991] and are summarized in Figure 1.

Each of the 98 subcollections was assigned to its own processor node, leaving 30 idle. Each of the nodes thus modeled a network server with the front-end machine playing the role of the broker and client interface. This

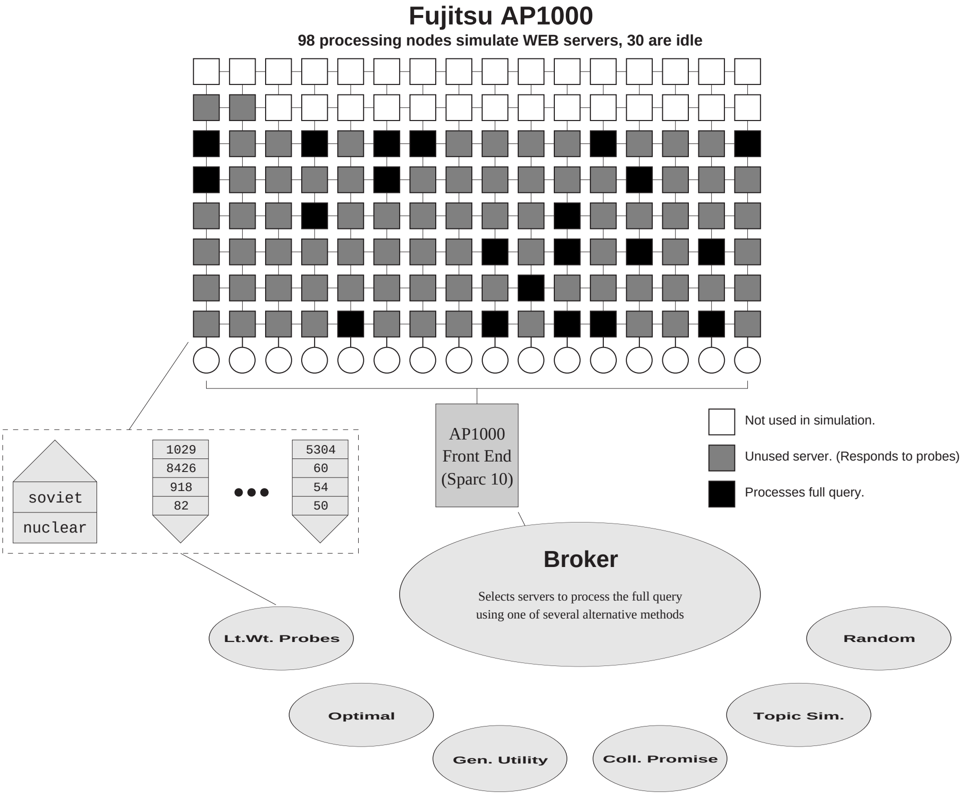
Fujitsu AP1000
98 processing nodes simulate WEB servers, 30 are idle
AP1000
Front End
(Sparc 10)
□ Not used in simulation.
▨ Unused server. (Responds to probes)
■ Processes full query.
soviet
nuclear
1029
8426
918
82
5304
60
54
50
Broker
Selects servers to process the full query
using one of several alternative methods
Lt.Wt. Probes
Random
Optimal
Topic Sim.
Gen. Utility
Coll. Promise

Fig. 1. Overview of the experimental model used throughout this article. The front-end of the Fujitsu AP1000 simulates the broker and the retrieval client interface. AP1000 nodes simulate network servers, each of which manages a subcollection of the overall text. In each experiment a fixed-sized subset F of available servers is used to process the full query. A variety of alternative methods for selecting F, represented as ellipses clustering around the broker, are compared. Some of them are based on knowledge derived from queries processed by all servers in the past. Another relies on transmission of a very cheap probe query to all nodes and reception of small reply packets of frequency data (as illustrated on the left of the diagram).

approach required no modification to PADRE's indexing structures which are naturally based on splitting up the data.

The use of a parallel machine was convenient rather than essential to the present study. Although the Fujitsu AP1000 architecture is logically a cluster of workstations and could potentially simulate a more realistic network (with random delays inserted to model real-world network latencies), the configuration available featured shared disks. Consequently, observed query-processing times on a node could not be used in the study, as they are not independent of activity on other nodes. A simplified cost model is described in Section 2.8.

PADRE mechanisms specifically implemented to support this study included

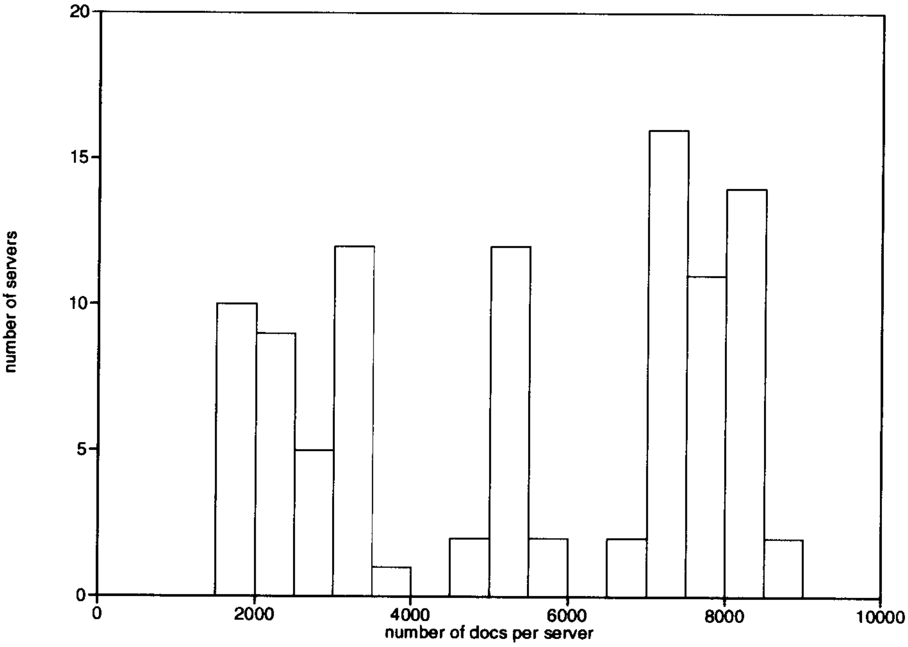

Fig. 2. Frequency distribution of number of documents per server.

(1) a mode in which local rather than global statistics are returned to the front-end and

(2) a command to select a subset of processors and to prevent nonmembers from contributing to the processing of a query.

2.3 Merging Rankings

It is assumed here that result merging is independent of the server selection method, though this may not always be the case.

"Theoretical" comparisons of server selection methods presented below assume perfect retrieval and perfect merging. However, server selection experiments using actual queries need to control for the potentially confounding effect of incorrect result merging. The difficulties inherent in merging rankings, even when the servers operate the same (*tf.idf*-based) algorithm, are described by Dumais [1992].

Hawking and Thistlewaite [1995] demonstrate that the use of a distance-based relevance scoring method enables correct merging, provided that all servers use it. Accordingly, empirical comparisons in the present study employ distance-based scoring to achieve correct merging and thereby allow direct comparison of server selection methods. Naturally, precision-recall performance would be expected to deteriorate if correct merging were not possible.

Note that the distance-based scoring is used only to achieve correct merging of results; the server selection methods examined do not rely on

```
topic 261
anyof "fissile |fissionable |plutonium |u235 |uranium "
anyof "armenia|azerbaijan|belorus|byelorus|c.i.s.
 |cis |estonia|gorbachev |kazak|kiev |kravchuk |latvia
 |lithuania|moscow |russia |russian |shevardnadzhe |siberia
 |soviet |u.s.s.r |ukrain|ussr |uzbek|yeltsin "
anyof "control|danger |lugar|safeguard |security |steal
 |stealing |stolen |theft |threat "
anyof "black market |terroris"
span  4
top 1000
```

Fig. 3. One of the distance-based queries used in the present experiments. The topic related to "dangers posed by fissionable materials in the states of the former Soviet Union." Each `anyof` creates a set of all match points for all terms in the given list of alternatives. The `span 4` scores the relevance of documents according to the spans they contain over the four match sets. An explanation of spans is given in the text and in cited references.

the relevance scoring method. A description of distance-based scoring appears in Section 2.4, and an example query is shown in Figure 3.

2.4 Research Topics and Queries

The research topics used in the experiments were topics 251–300 of the TREC set. Relevance judgments for all these topics are available for all of the data used.

A set of manually devised queries Q_{T5} for these topics was used in all the retrieval runs reported here. These queries were oriented toward good TREC performance and are consequently quite long. The average number of terms in each query was approximately 65. Long queries such as these are typical of high-performing queries in the TREC AdHoc tasks [Allan et al. 1995; Buckley et al. 1995]. An example query is shown in Figure 3.

Note that none of the server selection methods depend upon the manual generation of queries.

Relevance scoring of the queries was based on the method first described by Hawking and Thistlewaite [1995]. (Clarke et al. [1995] simultaneously reported an independently developed but nearly identical method.) These methods are based on lexical distance between term occurrences and are independent of collection statistics. They have been shown to be capable of achieving good precision-recall results on the TREC AdHoc tasks.

In processing the example query, each document is examined for spans of text including one term from each of the `anyof` lists. The fragment "`...terrorists using stolen plutonium from Kazakhstan...`" is an example of such a span.

The length of each span may be computed as the number of intervening nonquery terms (two in the example). The relevance score of the document which contains it is increased by an amount which depends upon some inverse function of span length. The rationale is that the fewer the number of intervening terms, the more likely that the query term occurrences are semantically related to each other. Hawking and Thistlewaite [1996] and Hawking et al. [1996] provide further explanation of the method and discuss the effectiveness of different functions of span length and the treatment of partial spans.

2.5 Query-Processing Data for Past Topics

The Topic Similarity and Server General Utility methods studied below rely on the availability of data about past query processing over all servers. TREC topics 202–250 were used as the past topics, but, unfortunately, relevance judgments were not available for these topics on CD4 documents. Accordingly, results of processing Q_{T4}, the best available set of PADRE queries for the past topics (a selection of the best of three independently generated sets), were used to estimate the missing information. Q_{T4} achieves an (unofficial) average precision of 0.3634 on the TREC-4 task. Of more relevance in this context, its R-precision (precision at the point when the number of documents retrieved equals the total number of relevant documents) was 0.4069. Q_{T4} queries use only distance-based relevance scoring.

2.6 Server Selection Effectiveness Measures

Server selection methods may be compared, for a given number $|F| = n$ of servers accessed, using

$$R_n = \frac{rel\ docs\ on\ F}{rel\ docs\ on\ S}.$$

This use of the R_n notation follows Lu et al. [1996].

Gravano and García-Molina [1996] criticize the use of measures, such as R_n, which are based on the location of relevant documents, pointing out that there is no benefit to the user in accessing a server containing relevant documents if the search engine there is unable to retrieve them. They propose measures $\mathcal{R}_n$ and $\mathcal{P}_n$ which are based on the ability of the server selection algorithm to predict which servers will return high-scoring documents.

Despite the above objection, the R_n measure has the considerable advantage that it is independent of the search engine(s) employed. Accordingly, R_n is used here. However, the performance of server selection methods in the context of a real retrieval system was also investigated, using measures based on relevant documents retrieved by the search engine in use. Search engines are normally compared on the basis of *recall* and *precision*. The proposed server selection effectiveness measures $\hat{R}_n$ and $\hat{P}_n$ report the proportions of all-server recall and all-server precision respectively which were obtained by accessing only n servers:

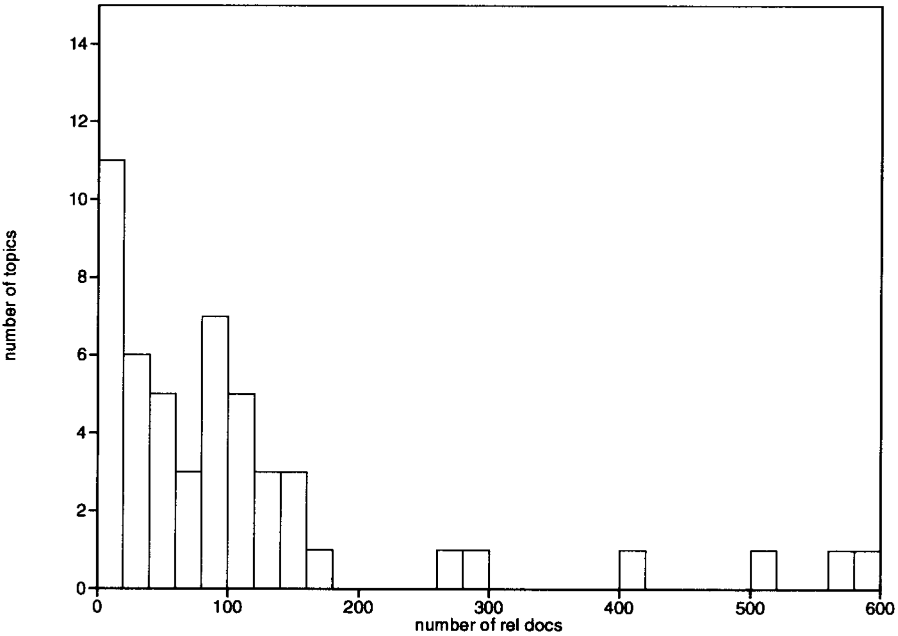

Fig. 4. Frequency distribution of number of relevant documents per topic.

$$\hat{R}_n = \frac{\textit{recall for } Q_{T5} \textit{ over } F}{\textit{recall for } Q_{T5} \textit{ over } S}$$

and

$$\hat{P}_n = \frac{\textit{precision@}20 \textit{ for } Q_{T5} \textit{ over } F}{\textit{precision@}20 \textit{ for } Q_{T5} \textit{ over } S}$$

R_n may be regarded as an *ideal* or *theoretical* form of $\hat{R}_n$. *Recall* is actually the number of relevant documents within the first 1000 retrieved. *Precision@20* is the number of relevant documents within the first 20 retrieved. The latter measure was chosen because it is a very real determinant of user satisfaction in the context of Internet searches.

Finally, an apparently satisfactory performance when averaged over 50 topics may conceal dismal failures on some topics. To measure this, the percentage of topics for which F held less than 10% of all the relevant documents is reported as the *failure rate*. Like R_n, this measure is independent of the search engine(s) employed.

2.6.1 *Relevant Set.* The relevant set for the experiments was defined as the set of documents judged relevant by the TREC assessors. The number of relevant documents per topic ranged from 1 to 594 and the mean was 110. Figure 4 shows the distribution of number of relevant documents per topic.

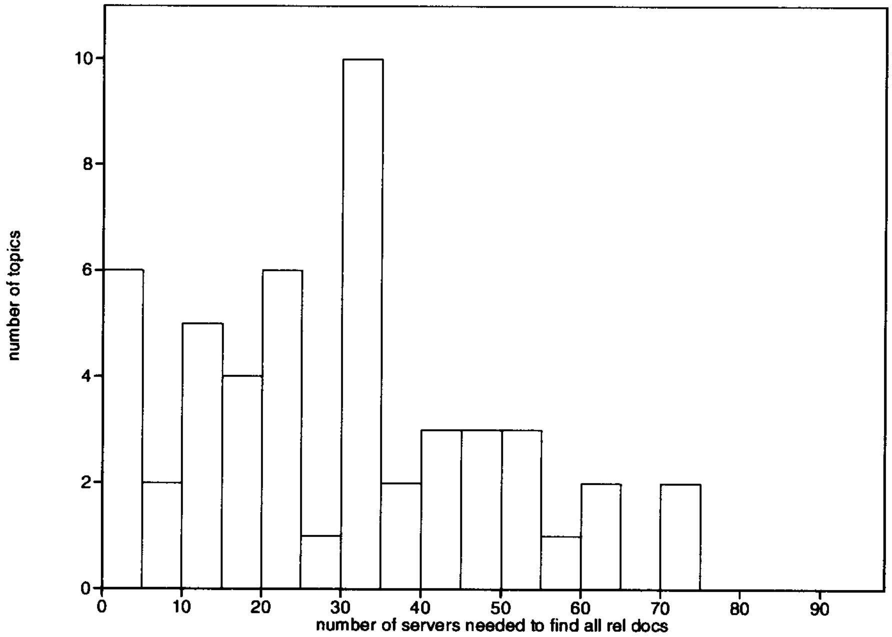

Fig. 5. Frequency distribution of number of servers holding relevant documents on a per-topic basis.

The distribution of relevant documents across servers was also determined and is shown in Figure 5. The number of servers holding relevant documents for a topic ranged from 1 to 71, and the mean was 29.

2.6.2 *Evaluation Framework.* A server ranking was evaluated by calculating R_n values and failure rates for $|F| = 1, \ldots, 98$. Then Q_{T5} queries were run over server subsets of size $|F| = 10,20,33,49,98$, and values of $\hat{R}_n$ and $\hat{P}_n$ were calculated.

When measuring the performance of a ranking method, the results are specific to the particular implementation of the method; a better implementation of the method will lead to better performance. For example, poor performance of an implementation of the Topic Similarity method may be due to poor choice of similarity metric rather than to a general deficiency of the method. This makes it difficult to make useful cross-method comparisons.

However, the availability of complete relevance judgments for the "new" topics makes it possible to estimate the maximum possible performance of a particular method on the stated task. Using the Topic Similarity example again, it is possible to rank past topics on the basis of the extent to which their distribution of relevant documents across servers matches that of the new topic, thus producing an estimated best-possible ranking of available past topics. Implementations of a method which rely on relevance judgments for new topics are totally impractical and are referred to here as "hindsight" implementations.

Results are presented for hindsight as well as for practical implementations of the methods considered here.

2.7 Incomplete Rankings

It was sometimes the case that less than $|F|$ servers achieved nonzero scores, either because the ranking method correctly assigned zero utility to useless servers or because the ranking procedure was deficient. It was assumed that the latter explanation was more often correct than the former, and, when necessary, additional servers were chosen by random selection to permit equal-cost (as measured by number of servers selected) comparison of methods. If zero scores are in fact due to a failure of the ranking method, this approach leads to fair comparisons of the methods.

However, as server selection methods approach the optimal, costs will tend to be overstated, particularly for large $|F|$. The issue of how to determine appropriate values for $|F|$ based on user and topic characteristics is discussed in Section 3.11.

2.8 Cost Model

Practical applications of distributed information retrieval may be subject to a variety of different cost models, depending upon the user, and the technological and economic circumstances in force at a particular time. Cost may be measured in terms of monetary charges levied by server, broker, and/or network operators, or in terms of service delays experienced by users and loads experienced by servers, brokers and networks.

The cost model used here is oriented toward monetary charges, but it is assumed that these may be derived directly from the computational and network resources used. This is almost certainly unrealistic, but in the absence of an accepted charging model, there is little practical alternative.

Under this model, the cost associated with distributed query processing depends upon the following:

(1) The number of servers accessed. It is simplistically assumed that the cost of processing an arbitrary query is equal for all servers. Thus the cost of processing a query over $|F|$ servers is $|F|/|S|$ of processing it over all servers. Actual query-processing times on the nodes of the parallel machine could not be used because the nodes share disks and thus do not function independently.

(2) The complexity of the query submitted to a server. Short queries are cheaper than long ones.

(3) The amount of data transferred between servers and the researcher's workstation. Transfer of documents for viewing is not included here, as it should be independent of the server selection method. Consequently, the majority of transmitted data which must be considered consists of lists of document identifiers (including scores, etc.).

Table I. Retrieval Performance of Random Server Ranking

Number of Servers	10	20	33	49
R_n	0.094	0.196	0.320	0.485
Failure rate	64%	16%	6%	0%

As yet, network latencies and timeouts have not been included. It would be necessary to include these factors if server selection comparisons were to be made on the basis of quality of service.

2.8.1 *Global Ranking.* In the actual retrieval runs, the local rankings at each server were actually merged using an efficient global reduction mechanism available on the parallel machine. In reality, the global-ranked list would be formed by sending local lists (or sublists) to the broker and merging them there. However, the results would be identical due to the use of distance-based scoring. The network traffic costs estimated in Section 3.10.3 assume the use of merging at the broker.

3. EVALUATION OF SERVER RANKING METHODS

Three of the experimental server selection methods—"Collection Promise," "Topic Similarity," and "Server General Utility"—rely on historical information derived from past query processing over all servers. In contrast, the "Lightweight Probes" method characterizes servers by requiring all of them to process a very short, low-cost query and return a packet of frequency information.

3.1 Controls

Two impractical methods—"Random" and "Optimal"—serve as controls.

3.1.1 *Random.* The method of randomly choosing a set of servers to process each query was used to verify that the simulation machinery was working and to establish a performance floor against which the other methods could be judged. Table I records the performance of the method. As can be seen, the proportion of relevant documents retrieved very closely approximates the proportion of servers used. The proportion of "failed" queries (as defined in Section 2.6) is quite high unless more than one-third of available servers are used.

3.1.2 *Optimal.* Given knowledge of the complete set of relevant documents, it is possible to choose a set of n servers which will achieve the best-possible recall of relevant documents for that value of n. Optimal runs are performed to serve as a performance ceiling.

Figure 6 illustrates the benefits to be gained if a close-to-optimal server ranking can be discovered in a real situation. On average, the best server for a topic holds nearly 11% of the relevant documents, and the best four hold approximately 30%. Table II reports the performance of the method on

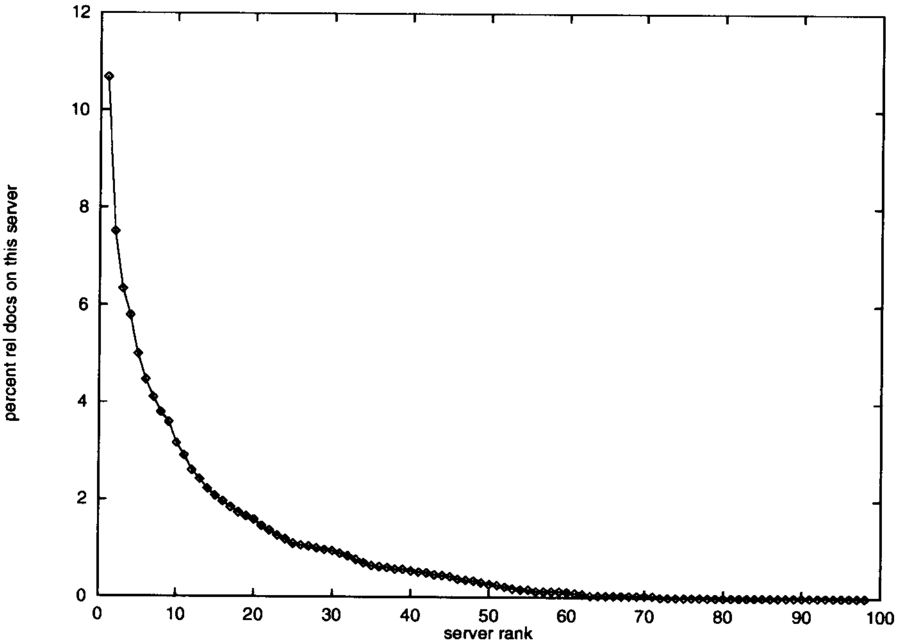

Fig. 6. The percentage of relevant documents for a topic held by servers as a function of their position in an optimal ranking. Data have been averaged over all 50 topics. Note that distributions for some individual topics depart considerably from this shape.

Table II. Retrieval Performance of the Optimal Server Ranking

Number of Servers	10	20	33	49
R_n	0.545	0.755	0.896	0.976
Failure rate	0%	0%	0%	0%

the chosen measures. Even when accessing only 10% of the servers, no queries "fail."

3.2 Server General Utility

This method assumes that some servers are better sources of relevant documents than others, regardless of the topic. Server utility was estimated with reference to the set of past topics (202–250). Because no past relevance judgments exist for these topics over CD4, the document ranks obtained using a full run with Q_{T4} queries were employed. If PADRE ranked a document rth among the documents retrieved by a Q_{T4} query, it contributed $1/r$ to the ranking weight of the server which held it.

The best-possible performance of this method on this task was estimated, using the complete set of relevance judgments over all new topics to compute the total counts for each server.

Table III and Figure 7 show the performance of this method as implemented compared with the estimated best-possible implementation. On

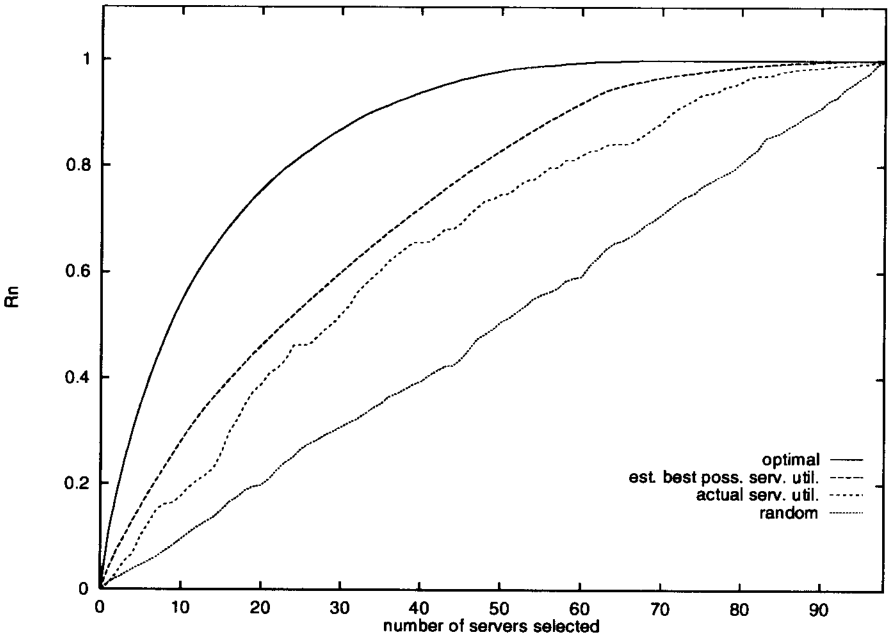

Fig. 7. R_n values for Server Utility methods compared with the controls.

Table III. Retrieval Performance of Server General Utility Ranking

Number of	Implemented				Estimated Best Possible			
Servers	10	20	33	49	10	20	33	49
R_n	0.177	0.388	0.577	0.739	0.281	0.461	0.638	0.820
Failure rate	38%	8%	0%	0%	18%	0%	0%	0%

both measures, results are considerably better than the Random method. Clearly some servers are generally more useful than others.

The estimated best-possible performance of the method is much inferior to the Optimal (only about half as good at 10 servers, using the R_n metric). This is not surprising as optimal rankings vary from topic to topic whereas the Server General Utility method takes no account of topic differences.

3.3 Collection Promise

This method assumes that subgroups of available servers serve particular categories of documents. Such an assumption is likely to apply in some real-world environments and is the case to a certain extent here. Collection promise could be estimated using a broad-brush version of the topic similarity methods discussed below. Instead, however, each new topic was manually assigned a list of collections (corresponding to single sources such as Wall Street Journal, Federal Register, etc.) considered most likely to supply documents in the area of the topic. If the target number of servers

Table IV. Retrieval Performance of the Collection Promise Method as Implemented, Compared with the Estimated Best-Possible Performance of the Method

Number of	Implemented				Estimated Best Possible			
Servers	10	20	33	49	10	20	33	49
R_n	0.202	0.282	0.412	0.549	0.280	0.448	0.622	0.780
Failure rate	38%	18%	10%	0%	4%	0%	0%	0%

was less than the total number of servers for the listed sources then a random sample of servers was drawn from each source.

Table IV records the results for this method. Using 10 servers, the method as implemented is much better than the Random method, but performance relative to the Random control declines as the number of servers increases. It is possible that the author of the collection orderings tended to do a reasonable job of choosing the most promising collection but a poor job of choosing the second and third best.

The same table also shows the estimated best-possible performance of the Collection Promise method on this task. These figures were obtained using knowledge of the server locations of all the judged relevant documents. Collections were ranked on a topic-by-topic basis according to ratio of relevant documents held per server. Results obtained are much better than those achieved by the human-assigned rankings and show a much lower percentage of failures. The actual and estimated best-possible Collection Promise rankings are compared with the controls in Figure 8.

3.4 Topic Similarity

This approach is derived from the Query Clustering (QC) and Modeling of Relevant Document Distributions (MRDD) methods of Voorhees et al. [1995].

Like General Utility, the Topic Similarity method assumes

(1) The existence of relevant historical data obtained by processing queries on ALL servers.

(2) That there is some sort of semantic relationship between the content of documents held by a single server. If documents are distributed across servers without regard to content, then this technique is unlikely to be useful.

As explained in Section 2.5 above, tables of the number of relevant documents per server for TREC topics 201–250 were used as the store of knowledge about past queries. Because no relevance judgments were available for these topics on CD4, Q_{T4} queries were first run over CD2 documents only. A relevance-score threshold was then set for each topic so that the number of documents above the threshold was equal to the number of

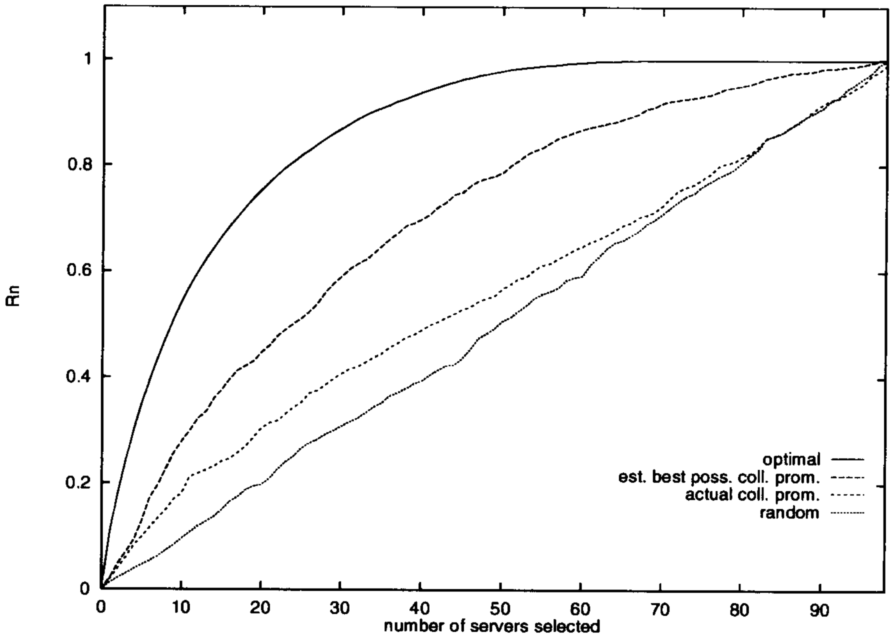

Fig. 8. R_n values for Collection Promise methods compared with the controls.

CD2 documents actually judged relevant. The Q_{T4} queries were then run over CD2 and CD4 together and documents exceeding the threshold score for the topic were assumed to be relevant. The number of "relevant" documents held by each server for each past topic was then tabulated. Note that the use of an absolute score cutoff here is justified by the use of distance-based scoring. A document's score may be determined in isolation from any collection.

When generating server subsets for a new topic, the distributions of relevant documents across servers for each of the similar past topics were accumulated, and the resulting server-indexed array was sorted in descending order of number of relevant documents, giving a server ranking.

Two alternative methods of topic similarity computation were used: *manual* and *automatic*. In the manual method, a list of past topics considered most likely to involve documents in the same subcollection was manually derived by scanning the set of topics. The maximum number of similar past topics assigned to any one new topic was six, and the mean number was 3.7.

The automatic method used the SMART [Buckley et al. 1996] retrieval system to compute vector-space similarities. In the latter computations, `relevant`, `document`, and `contain` were added to the stopword list, and *idf* values derived from the full two-gigabyte data set were used in generating weights. It is not unreasonable to do this, since the method assumes past access to data on all servers. A straight cosine match with *tf*idf* variant was used (SMART `ltc`).

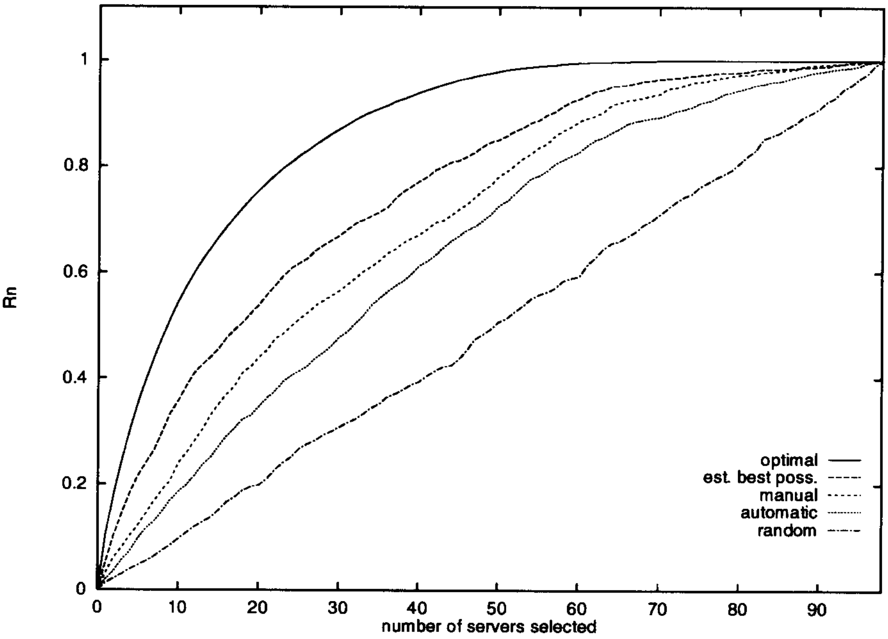

Fig. 9. R_n values for Topic Similarity methods compared with the controls.

Table V. Retrieval Performance of the Automatic and Manual Topic Similarity Implementations Compared with Estimated Best Possible Performance of Any Topic Similarity Implementation. The automatic version used an average of 5.9 similar topics.

Number of	Automatic				Manual				Estimated Best Possible			
Servers	10	20	33	49	10	20	33	49	10	20	33	49
R_n	0.181	0.335	0.512	0.702	0.199	0.368	0.538	0.755	0.399	0.571	0.726	0.864
Failure rate	28%	8%	0%	0%	16%	8%	4%	2%	0%	0%	0%	0%

A maximum of six most similar past topics were chosen (average 5.9), and these were used to generate results reported below. A run based on a smaller average number (matching that of the manually generated similarities) of similar past topics yielded slightly poorer performance on all measures.

3.4.1 *Results for Topic Similarity Methods.* An estimate of the best-possible performance of Topic Similarity methods on this particular problem was made by creating a normalized vector, for both new and past topics, of the number of relevant documents per server. For each new topic, inner products were computed between its vector and those of each of the past topics, allowing past topics to be ranked by "similarity" to the new topic. Heuristics were used to choose the number of similar topics in each case and resulted in a maximum of six and an average of 3.7. Better heuristics may result in better performance. Table V and Figure 9 show that excellent results on this task were obtainable using hindsight-generated topic similarities. The proportion of relevant documents obtained from 10 servers was four times higher than that from Random ranking. No queries failed.

Unfortunately, neither manual nor automatic assignment of similarities approached the performance level of the hindsight method. However, results for both categories are much better than random, with manual outperforming automatic.

3.5 Lightweight Probes

Experience in TREC has shown that high precision-recall performance requires complex queries; most of the best-performing systems make use of large-scale query expansion. However, complex queries take much longer to process than short queries, raising the question of whether very short, efficiently processed "queries" could be used to select servers to process the full query.

The Lightweight Probe method proposed here (believed novel) broadcasts a small number p of terms to all available servers, each of which responds with a small packet of term frequency information. The frequency data are then used to rank the likely utility of the servers.

There are some similarities between the Lightweight Probe method and the facilities provided within the Stanford Protocol Proposal for Internet Retrieval and Search (STARTS) [Gravano et al. 1997] for extracting metadata and content summaries from servers. However, in the STARTS proposal it is envisaged that full-content summaries are obtained from servers "periodically." By contrast, full content summaries are never obtained in the Lightweight Probe method. Instead, each query is preceded by a request for a minimal amount of such information.

The Lightweight Probe approach assumes that

(1) tiny probes can be processed with a significant cost saving over a full query;

(2) useful information about the holdings of a server may be deduced using a low-cost probe; and

(3) probe bandwidth and latency are low.

It should be noted that a p-term probe is likely to be significantly cheaper to process on a server than a p-term query, because no document ranking is required. In all experiments reported below, $p = 2$, and the frequency information returned by the servers comprised

(1) D_i: the total number of documents on the server,

(2) f_{prox}: the number of documents containing a specified number of the terms within a specified proximity of each other,

(3) $f_{cooccur}$: the number of documents in which a specified number of the terms cooccur,

(4) f_i: the number of documents containing each individual term t_i.

These definitions of probes and probe replies were chosen to keep probe processing and network costs low while providing the broker with sufficient information to enable good ranking. Note that, to permit stemming of probe terms, all servers must use the same stemming algorithm.

3.5.1 *Formula for Combining Probe Frequency Data.* The frequency data returned by probes may be combined in many different ways in order to produce a set of server scores. A number of formulae and many combinations of weighting coefficients were tested. However, to reduce the extent to which the methods were too closely tuned to the particular problem characteristics, training was done only on topics 251–275, using the manually generated probes. The best formula found during training was used in deriving server rankings for both the automatic and the manual runs reported below.

In the chosen formula, raw frequencies were converted to proportions of the total frequency in order to avoid an undesirable bias toward higher-frequency terms. However, some terms are more useful than others, and the formula includes utility weightings to reflect this:

$$S_i = c_1 f_1' + c_2 f_2' + c_3 f_{\text{cooccur}}' + c_4 f_{\text{prox}}'$$

The primes indicate the use of normalized rather than raw frequencies.

In training, best results were obtained with $c_4 = 100$, $c_3 = 10$, $0 \le c_2 \le 1$, and $0 \le c_1 \le 1$. Actual values of c_1 and c_2 were derived from the summed raw frequencies as follows:

$$c_i = \begin{cases} \sum_{j=1}^{|S|} f_{ij}/f_t & \text{if } \sum_{j=1}^{|S|} f_{ij} <= f_t \\ f_t / \sum_{j=1}^{|S|} f_{ij} & \text{otherwise} \end{cases} \tag{1}$$

where f_{ij} is the raw frequency of probe term i on server j. This function is represented graphically in Figure 10. Terms were thus weighted according to how closely their total frequency of occurrence approximated a target total frequency f_t. The target frequency used was 110, which was the average number of relevant documents per topic. The rationale for this weighting was that terms with around the target frequency were expected to be more useful discriminators than those with very high or very low document frequencies.

3.5.2 *Manual Probe Generation.* Manual selection of a set of probe terms required less than one minute of human time per topic. The approach taken to selecting terms for this purpose was quite different to normal manual query generation; terms were selected which were reasonably specific to the general subject area of the topic and likely to occur reason-

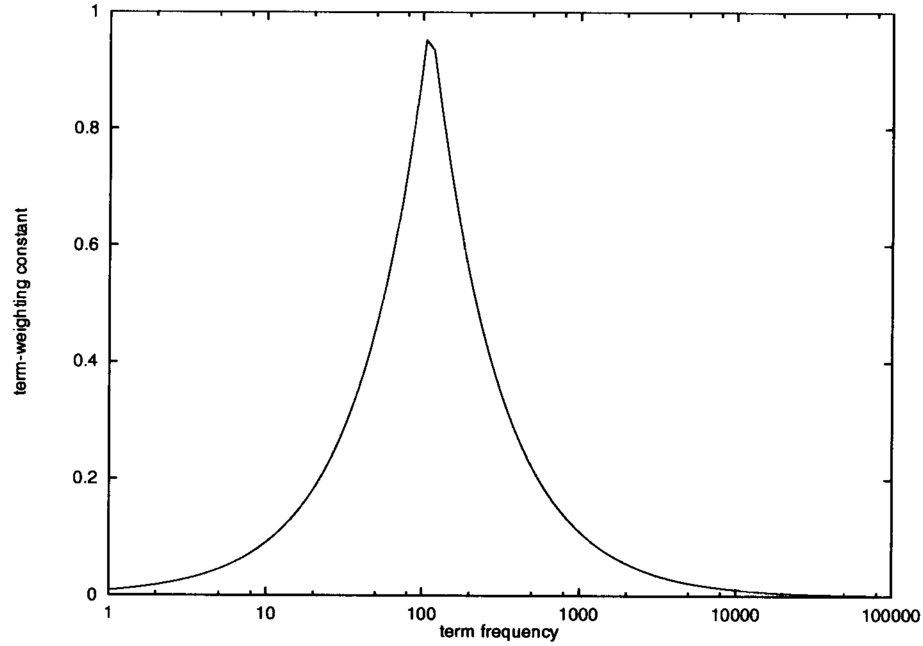

Fig. 10. Graphical representation of the individual probe-term weighting function given in Eq. (1). Note that the horizontal scale is logarithmic.

ably frequently within documents in that area. For the topic *Exportation of US Industry*, probe terms were `offshore` and `manufactur`, neither of which occurred in the topic statement. Extensive use was made of stems, in order to register occurrences of as many morphological variants as possible of the chosen terms.

3.5.3 *Automatic Probe Generation.* There are many possible ways in which probe terms may be automatically selected. The present experiments used stemmed nonstopword terms extracted from the topic title and converted to lower case. Where there was only one term in the title, that term was repeated. Where there were more than two terms, the two which occurred least frequently in a reference collection (described below) were chosen.

3.5.4 *Document Frequency Reference Collection.* In selecting probe terms on the basis of document-frequency (*df*) information, it would clearly be cheating to obtain these frequencies from the distributed test data. However, use of a table of document frequencies extracted from a static reference collection is considered realistic and is expected to produce reasonable rankings of probe terms most of the time.

Accordingly, a reference collection was created by sampling every eighth file from TREC CDs 1–4, resulting in a collection of 178,093 documents totalling 554 megabytes of text. The reference collection thus included

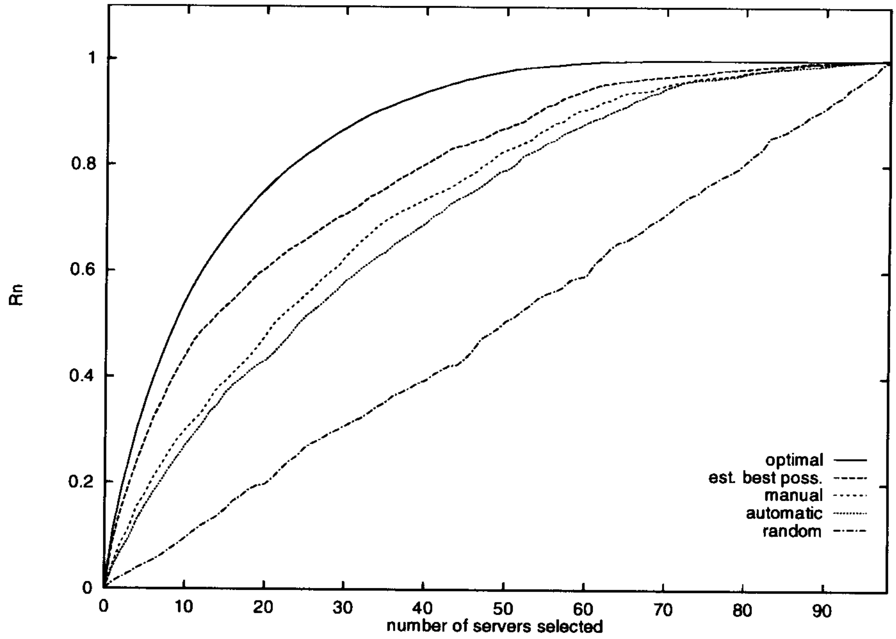

Fig. 11. R_n values for Lightweight Probe methods compared with the controls.

Table VI. Retrieval Performance of Automatically and Manually Generated Lightweight Probes Compared with Estimated Best-Possible Performance of Probe Words Selected from those in Topic Description and Narrative

Number of	Automatic				Manual				Estimated Best Possible			
Servers	10	20	33	49	10	20	33	49	10	20	33	49
R_n	0.271	0.431	0.616	0.787	0.300	0.478	0.670	0.822	0.441	0.604	0.741	0.867
Failure rate	14%	2%	0%	0%	20%	2%	0%	0%	0%	0%	0%	0%

approximately one-eighth of the test data plus a similar amount of data not used in the test.

The size of the reference collection is unlikely to be important. In particular, there should be no necessity to increase its size in response to increases in overall data size. All that is required is a rough approximation to the term frequency distribution of the real data.

3.5.5 *Estimated Upper Limit to Performance of Lightweight Probes.* Obtaining a firm upper limit on the performance of two-term probes would be prohibitively expensive, as there are of the order of 5×10^5 distinct words in the test data and thus of the order of 10^{11} distinct (unordered) pairs of terms, without even considering stems or phrases.

To reduce the computational load to something more manageable, only words occurring in the topic specification were considered. This list was further filtered to remove stopwords. For each topic, a normalized vector of

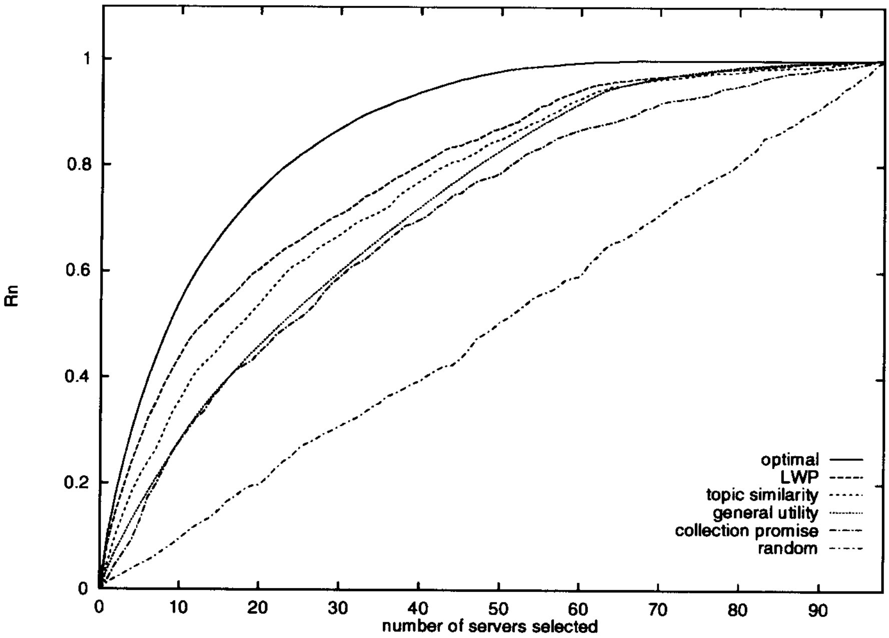

Fig. 12. R_n values for estimated best-possible rankings for each different method.

relevant documents per server was compared with a normalized vector of server scores for each distinct probe term pair. The probe term pair giving rise to the highest inner product was considered to be the best.

3.5.6 *Lightweight Probe Results.* As can be seen from Table VI and Figure 11 the limit to performance of Lightweight Probes is surprisingly high, even when probe terms are restricted to words occurring in the topic specification. Even better performance is likely to be achievable if this restriction were to be removed, allowing arbitrary words, phrases, stems and larger numbers of terms.

Server ranking using results from manually devised probes performed well. Performance of the automatically generated probes was marginally worse but still at a practically useful level.

It is interesting to compare the probe terms selected by automatic and manual with those selected in the exhaustive evaluation described in Section 3.5.5 above. As may be seen from Table VII, there is relatively little overlap between term pairs derived using the three different methods. Some of the term pairs in the "Best Probes" column (such as `long/those`, `elephant/if`, and `being/law` seem unlikely to be chosen by any imaginable general probe selection method. However, examination of full probe rankings has revealed that more intuitively acceptable pairs (for example, `colleges/philosophy`, `ban/population`, and `drug/information` respectively) score nearly as highly.

Table VII. Comparison of the Pairs of Probe Terms Derived by Three Different Means. Note that the terms in the best-possible column were treated as whole words whereas those in the automatic and manual columns were treated as word prefixes except for `heart` in topic 254.

Topic	Title (Sometimes Abbreviated)	Manual Probes	Automatic Probes	Best Probes
251	Exportation of Industry	offshore manufactur	export industr	country lost
252	Combating Alien Smuggling	illegal immigrant	smuggl alien	stop describe
253	Cryonic suspension services	cryo human	cryonic suspens	cryonics storage
254	Non-invasive heart procedures	heart treat	ailment invas	heart persons
255	Environmental Protection	environment conservation	protect environment	environmental resources
256	Reduced Core Curriculum	educat curricul	reaction neg	long those
257	Cigarette Consumption	smok cigar	cigarett consumpt	cigarette causative
258	Computer Security	comput secur	securit comput	networks technology
259	New Kennedy Assassination Theories	Kennedy assassin	theori assassin	assassination must
260	Evidence of human life	archaeolog prehist	evid human	ago indicating
261	Threat posed by Fissionable Material	nuclear soviet	fission pose	plutonium underpaid
262	Seasonal affective disorder syndrome	SADS day	syndrom disord	seasonal disorder
263	Algae as Food Supplement	alga supplement	alga supplem	algae blue
264	U.S. Citizens in Foreign Jails	citizen prison	jail citizen	if intervention
265	Domestic Violence	domestic violen	violenc domest	domestic u.s.
266	Professional Scuba Diving	div scuba	scuba dive	professionally work
267	Firefighter Training	fire train	firefight train	changing skills
268	Cost of national defense	militar budget	defens cost	ground submarine
269	Foreign Trade	trad polic	foreign trade	policy order
270	Control of Food Supplements	supplement federal	supplem food	supplements supplement
271	Solar Power	solar energy	solar power	energy extensively
272	Outpatient Surgery	patient hospital	outpati surger	outpatient insurance
273	Volcanic and Seismic activity	volcan earthquake	volcan seismic	there activity
274	Electric Automobiles	electric vehicle	automobil electr	electric operation
275	Herbal Food Supplements/Natural Health	herb health	herbal supplem	medicines health
276	Imposition of a school uniform	school uniform	imposit dress	gains achievement
277	Civilian Deaths from Land Mines	mine civilian	civilian mine	land cessation
278	DNA Information about Human Ancestry	DNA gene	ancestr dna	discuss mankind
279	Earth magnetic pole shifting	magnet pole	pole magnet	degree well
280	Ban on Ivory Trade	ivory elephant	ivor trade	elephant if
281	Genetic code of yeast	gene yeast	yeast genet	identify genetic
282	Violent Juvenile Crime	crim juvenile	juvenil violent	juvenile rape
283	China Trade	chin trad	china trade	chinese brought
284	International Drug Enforcement	drug agen	enforc cooper	being law
285	World submarine forces	sub defen	submarin forc	nuclear suffering
286	Paper Cost	paper cost	paper cost	rising processing
287	Electronic Surveillance	surveillance employ	surveill electron	privacy surveillance
288	Weight Control/Diets	weight diet	diet weight	weight people
289	For-profit hospitals	hospital profit	hospit profit	consumer concentration
290	Foreign automobile makers in U.S.	auto factor	automobil manufactur	foreign locations
291	Source of taxes	tax revenue	tax sourc	tax evidence
292	Worldwide Welfare	welfare gov	welfar worldwid	countries source
293	Evacuation of U.S. Citizens	civil evacuat	evacu citizen	u.s. non
294	Husbandry for exotic animals	farm rais	husbandr exot	out discover
295	Deaths from Scuba Diving	scuba div	scuba dive	diving occurred
296	Trash TV	television entertain	trash tv	trash usa
297	Right To Die - Pros and Cons	euthanasia die	die right	years reports
298	Gun Control	gun crime	gun control	gun measures
299	Impact of military downsizing	defense base	downsiz militar	base foreign
300	Air Traffic Control Systems	air crash	traffic air	u.s. association

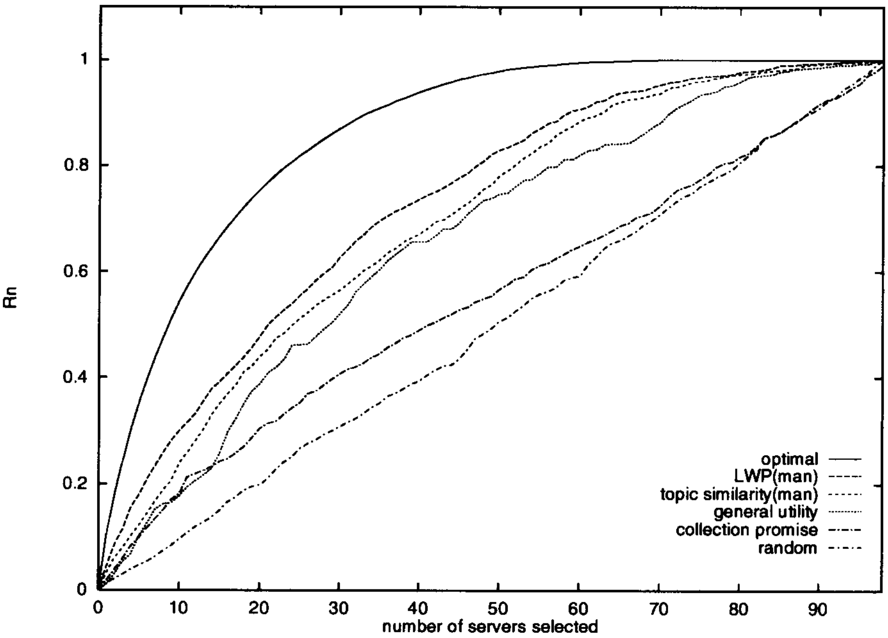

Fig. 13. R_n values for the best rankings for each method, achieved without the aid of hindsight.

3.6 Comparison of Estimated Best Possible Performance of the Methods

Figure 12 compares the estimated best-possible performance of the four experimental methods with the performance of the controls. The following observations may be made:

—All experimental methods are capable of performing significantly better on this task than Random ranking.

—None of the experimental methods investigated appear to be capable of generating optimal rankings.

—Lightweight Probes show somewhat more potential than other methods while Collection Promise and Server General Utility show somewhat less.

—From Tables III, IV, and VII, it may be seen that all methods are potentially capable of avoiding failures on this task, provided at least 20 servers are used. However, even the idealized forms of Server General Utility and Collection Promise experienced failures at the ten-server level.

3.7 Comparison of Practical Implementations of the Experimental Methods

Figures 13, 14, and 15 compare the best performance, achieved in the absence of hindsight, of the four experimental methods with Random and Optimal rankings. The following observations may be made:

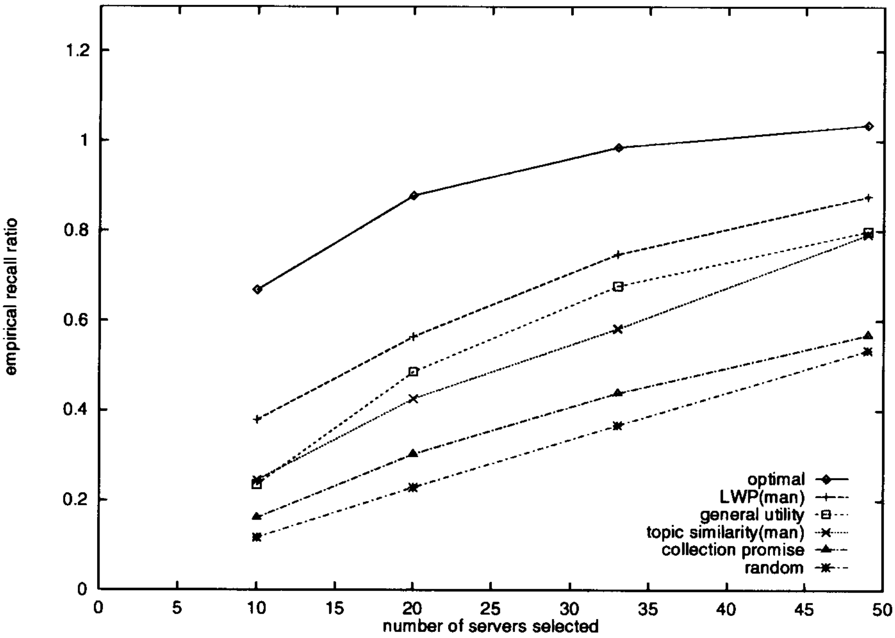

Fig. 14. $\hat{R}_n$ (empirical recall ratio) values for the best rankings for each method, achieved without the aid of hindsight. These are ratios of recall achieved by the subset F to the recall achieved over the full set of servers S. As noted in the text, recall is measured at 1,000 documents retrieved and the ratio may exceed unity.

—Given imperfect queries such as Q_{T5} it is possible to achieve better precision and recall results from a subset of servers than from all of them, as witnessed by the greater-than-unity values for both $\hat{R}_n$ and $\hat{P}_n$ when using an Optimal server ranking. A slightly greater than unity $\hat{P}_n$ value was also obtained for manual Lightweight Probes when using 49 servers.

—Close inspection of Figures 13 and 14 suggests that many, if not all, of the $\hat{R}_n$ values exceeded the corresponding R_n values.

—All experimental methods performed better than the Random method.

—All experimental methods performed much worse than the Optimal method.

—Collection Promise was the worst of the practical methods.

—Topic Similarity was clearly superior to Server General Utility on the theoretical measure, but there was no consistent difference between them on the empirical measures.

—Manually generated Lightweight Probes achieved results which were superior on all three measures to those of any other experimental method.

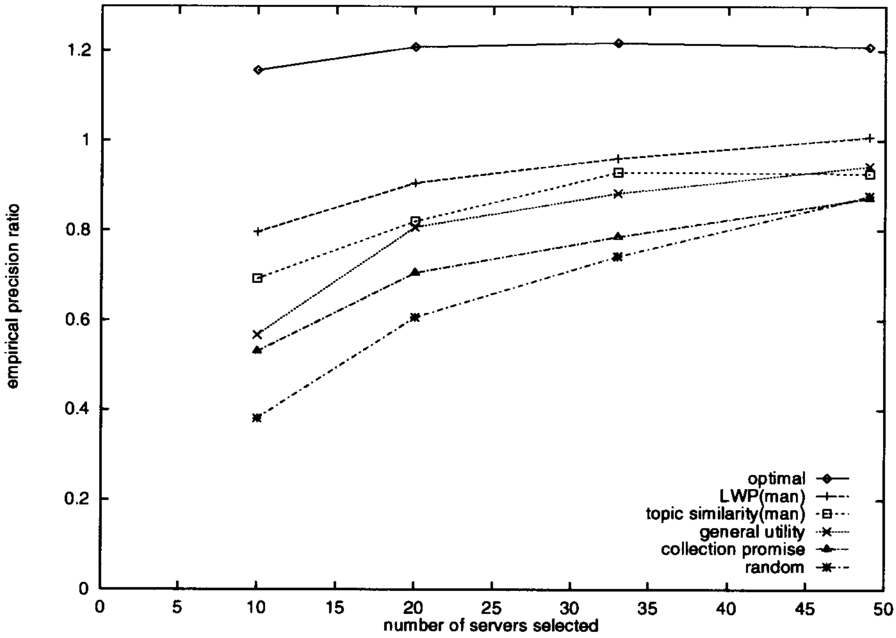

Fig. 15. $\hat{P}_n$ (empirical precision ratio) values for the best rankings for each method, achieved without the aid of hindsight. These are ratios of precision achieved by the subset F to the precision achieved over the full set of servers S. As noted in the text, the ratio may exceed unity.

3.8 Performance of Automatic Methods

The performance of automatic rankings is of much greater practical relevance than that of manual or hindsight implementations. Figure 16 compares the performance of the various automatic methods. All three methods performed better than the Random control. The performance of Lightweight Probes was consistently superior to the other two.

3.9 Confirmation of Generality of Automatic Lightweight Probe Results

In order to confirm that the encouraging results achieved by the automatic Lightweight Probes method were capable of generalizing to other data and other topics, a follow-up experiment was conducted, as follows:

—TREC CDs 1 and 2 were arbitrarily distributed across 98 simulated servers. As with the data distribution in the main experiment, servers were assigned data from only one collection, and the amount of data (in bytes) was comparable across servers.

—Lightweight probes were generated for 199 of the 200 TREC topics for which relevance judgments were available over this data set (topics 1–3 and 5–200). The Probe Generation method was exactly the same as in the main automatic probe experiment and used the same reference collection for ranking the title terms.

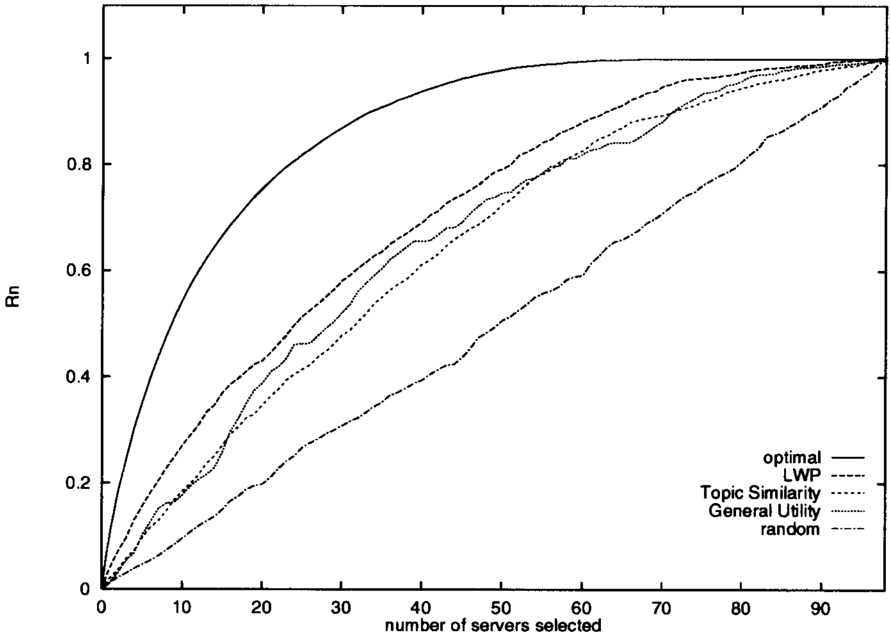

Fig. 16. R_n values for automatically generated rankings.

—R_n values and failure rates were calculated based on the known relevant document locations.

This follow-up experiment used a different data set and three times as many topics as the main experiments. Unfortunately, two minor problems were subsequently discovered with the data used. First, a formatting inconsistency in the title of topic 4 caused a failure of probe generation and necessitated exclusion of that topic. Second, formatting problems with a small number of document identifiers resulted in the exclusion of 17 relevant documents from the total of 56,200. The effect of this problem is clearly negligible, even in the worst case.

Results presented in Tables VIII and IX and Figure 17 show that the Lightweight Probe method performed at least as well here as in the main experiment, thus confirming that the success of the method was neither specific to the particular conditions of the main experiment nor due to any training effect.

3.10 Costs

As stated in Section 2.8, the principal orientation of the cost model employed here is toward financial charges levied by server and network operators. These are assumed to be simply derived from actual computational and network costs.

3.10.1 *Broker Costs in Generating a Server Ranking.* The computational costs incurred by the Broker in generating server rankings are unlikely to

Table VIII. Comparison of Results of Automatically Generated Lightweight Probes with Random and Optimal Controls Using TREC Topics 1–200 over TREC CD1 and CD2

Number of	Random				Automatic LWP				Optimal			
Servers	10	20	33	49	10	20	33	49	10	20	33	49
R_n	0.100	0.208	0.345	0.511	0.297	0.500	0.675	0.817	0.500	0.730	0.889	0.973
Failure rate	56%	31%	16%	9%	12%	3%	0%	0%	0%	0%	0%	0%

Table IX. Comparison of Results of Automatically Generated Lightweight Probes Obtained Using TREC Topics 1–200 over TREC CD1 and CD2 with those Obtained Using TREC Topics 251–300 over TREC CD2 and CD4

Number of	CD2/4, Topics 1–200				CD1/2, Topics 1–200			
Servers	10	20	33	49	10	20	33	49
R_n	0.271	0.431	0.616	0.787	0.297	0.500	0.675	0.817
Failure rate	14%	2%	0%	0%	12%	3%	0%	0%

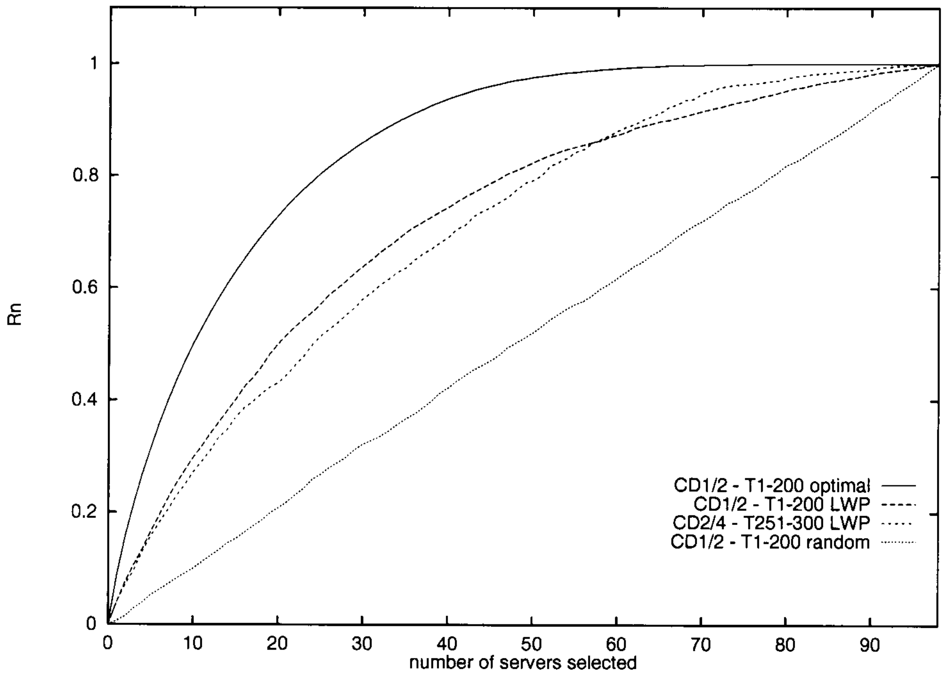

Fig. 17. Comparison of results of automatically generated Lightweight Probes obtained using TREC topics 1–200 over TREC CD1 and CD2 with those obtained using TREC topics 251–300 over TREC CD2 and CD4.

be significant. Server General Utility costs are obviously low because the ranking is effectively fixed. Automatic Topic Similarity methods require vector-space similarity computations between the new topic and each of the library of past ones. Automatic Lightweight Probes currently require only

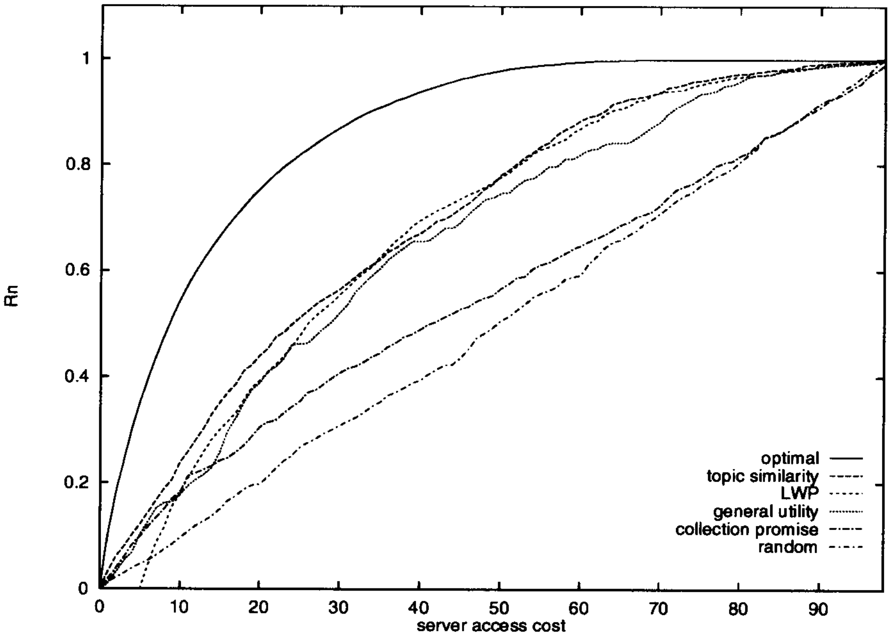

Fig. 18. R_n values for best no-hindsight rankings plotted against the server access cost where one unit of cost is incurred by processing a full query on one server.

extraction of title terms and determination of their frequencies in the reference collection, which may involve only a table lookup.

3.10.2 *Server Access Costs.* All methods include a core server-access cost component of $|F|/98$ of the cost of processing the query on all servers. The Lightweight Probe method (only) incurs an additional cost for processing the probes. The ratio of the average time taken to process a full query to the average time taken to process a two-term probe of this form was observed to be **20.8**. This ratio was computed using actual elapsed times for PADRE runs over all 98 nodes (servers).

Using the above ratio and assuming uniform server costs proportional to the computational load imposed, the aggregated server cost of processing the probe is estimated to be a little less than that of running the full query over five servers. In other words, to achieve an equal-cost comparison with other nonprobe methods, the number of servers processing the full query in the Lightweight Probe case should be reduced by five. Figure 18 shows performance of the alternative methods against server-access costs. As may be observed, even at 10 units of cost, the Lightweight Probe method matches the performance of Server General Utility and Collection Promise. By 33 cost units, it matches or surpasses the performance of all other practical methods.

3.10.3 *Network Costs.* The length of queries used in the experiment averaged a little over 800 bytes, and document descriptors returned in response to queries are likely to be approximately 200 bytes in length (title, sample text, and document identifier (URL)). Probes were an average of 14 bytes long, and probe responses were each 20 bytes.

The network cost of transmitting document descriptors depends upon the precise algorithm used for combining rankings and upon the particular pattern of scores. The number of document identifiers transmitted can be minimized if each server in F sends only its top-ranked item to the broker and further items only on demand. In this case the number of unnecessary document identifiers sent would be no more than $|F| - 1$. However, such an algorithm would be severely affected by network delays. More realistically, servers would transmit buffers of k document identifiers, in which case the excess number sent could be as high as $k \times |F| - 1$ in the worst case. Minimax techniques could be used to reduce unnecessary traffic.

The additional network traffic resulting from probes and probe responses in the experiments reported above was thus $98 \times 34 = 3.3 \times 10^3$ bytes. Assuming $k = 10$, this compares with a total of 2.8×10^3 bytes for each server which receives the full query and transmits a packet of k document descriptors. The total network traffic cost of probe processing is thus equivalent to that of processing the full query on less than 2 servers and may generally be neglected. Probe responses may be reduced to 4 bytes or less by preevaluating the server score prior to transmission. In reality, however, network packet headers are likely to add at least 16 bytes to each message, resulting in a net increase in the relative network cost of using probes.

The issue of network latency and timeouts is discussed in Section 4.

3.11 How Many Servers to Use

The retrieval runs reported above were subject to fixed values of $|F|$ but scant consideration has been given to determining the appropriate value of $|F|$ or determining how to vary $|F|$ from topic to topic. The best stopping point will probably vary from topic to topic and must depend upon a characterization of the requirements of the user. Significant user characteristics may include

(1) the user's assessment of the value of the first relevant document retrieved relative to the cost of a server access and

(2) the decrease in assessed value of relevant documents as more and more are found.

The latter is certain to depend in practice upon how similar a new relevant document is to ones which have been retrieved earlier.

User characteristics may vary enormously depending upon the purpose for which documents are being retrieved.

For purposes of illustration, let the cost of accessing a server (including network costs) be \$1.00 and consider two hypothetical users, U1 and U2. U1 assigns a value of \$0.60 to retrieving any document, whereas U2 values

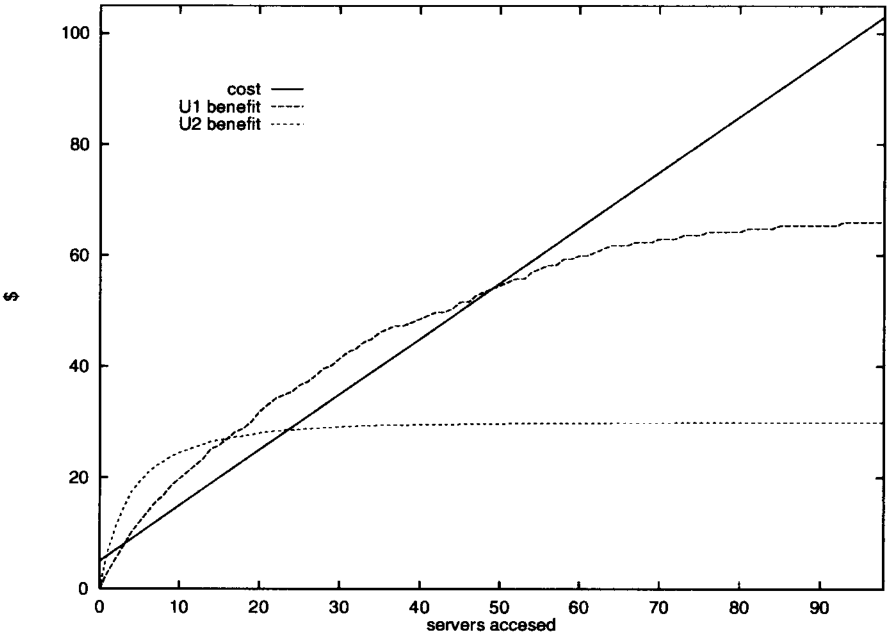

Fig. 19. Costs and benefits for two hypothetical users plotted against number of servers accessed. The benefits are calculated using the manual Lightweight Probe method and assuming an ideal retrieval engine on each node.

the first relevant document at $1.50 and each subsequent document at 0.95 of the value of the one immediately preceding it. Figure 19 plots net benefit gained for the two users and cost against number of servers accessed using the manual Lightweight Probe method to select servers. Using the second cross-over between the benefit and the cost lines as the stopping point indicates that U1 should access about 50 servers and U2 only 25. Note, however, that these figures are averaged over 50 topics.

It has been mentioned earlier that ranking methods sometimes assign zero scores to significant numbers of servers. The number of nonzero (or above some threshold) scores may suggest a natural size of *F*. Alternatively, the clustering approach of Callan et al. [1995] may be used.

4. DISCUSSION AND CONCLUSIONS

The experimental environment represented by the TREC-5 Database Merging task and the TREC relevance judgments has provided a very valuable opportunity to test server ranking methods in a way which would be impractical using real servers on the Internet. This environment has allowed discovery of optimal rankings, estimation of performance limits to methods, and accurate comparison of alternative methods.

However, as always, caution must be exercised when applying laboratory results in the field. Differences in numbers of servers, in data distribution across servers, and in types of documents found in a real situation may affect the applicability of results reported here. Furthermore, in an interactive distributed-search situation, the evaluation of document relevance against a static statement of information need, as performed here, is unrealistic.

The lack of official relevance judgments for past topics over half of the data is likely to have limited the performance of the implemented methods which relied on historical data, that is, Topic Similarity and Server General Utility. However, such judgments are very unlikely to be available in real-world network server applications, and it could be argued that the use of human relevance judgments at all was unrealistic. Furthermore, the Q_{T4} queries are probably unrealistically good, having been tuned (by selection among alternatives) on the basis of official relevance judgments.

In the present experiments, a very simple model of network costs was used, but the experimental framework allows for future simulation of more sophisticated networking models. It is clear that network latencies and the necessity to impose timeouts will be critical to the successful adoption of Lightweight Probes on the Internet or on large-scale corporate networks. The costs and benefits of the Lightweight Probe method in such environments will depend not only upon the distribution of latencies but also on network and server reliability as well as on the charging regimes of the available servers. It is not envisaged that Lightweight Probes would be broadcast to all servers on the Internet but only to those on a list of servers maintained by the Broker (perhaps supplied by the searcher's organization).

Determination of an optimum value of $|F|$ will require a model of the user's requirements and a characterization of the topic.

It is not considered worthwhile to undertake further investigation of the Collection Promise method. The obvious automatic methods for listing appropriate collections such as topic similarity and content characterization are almost certainly better applied on a server rather than a collection basis.

The relevance of server ranking methods relying on historical data to real environments is questionable due to the dynamic nature of server holdings and to the low likelihood that good past data will be available for even static servers.

The relatively good performance of Server General Utility ranking is notable, as it should be less affected by noncataclysmic changes in server contents than other historical data methods. The good performance may be an artefact of the TREC data and topics, but intuition suggests that human interests and the practicalities of server holdings will re-create the phenomenon to an even greater degree in the real world.

It would be interesting to evaluate other methods such as those of Gravano and García-Molina [1996], Callan et al. [1995], Dolin et al. [1996], and Chakravarthy and Haase [1995] in the experimental environment used here. If the server characterization methods of Dolin et al. [1996] and Chakravarthy and Haase [1995] can be performed automatically and effectively (preferably by the server itself), their methods would be potentially applicable to dynamic server environments.

Although the Lightweight Probe method attracts additional network and server access costs, it is not subject to the limitations of historical data methods. It is thus very fortunate that potential performance of the method is comparable with that of the other methods, even after compensating for additional costs. The relatively good performance of the simple automatic method for generating probe terms is encouraging, and it seems likely that among the very large space of possible automatic methods will be found a way of at least equalling the current manual performance.

It should be remembered that the Lightweight Probes method is only applicable when the cost of processing them is significantly less than that of processing full queries. Whether this is the case may depend more upon the fee structure of server and network usage than on actual computational cost.

A number of additional questions are raised by the work on Lightweight Probe queries:

(1) Is it worthwhile to consider larger values of p, or even $p = 1$? Given that costs will increase with increasing p, the answer may depend upon the likely average size of $|F|$.

(2) Should occurrence frequencies rather than document frequencies be used for the f_i?

(3) Is the current probe-scoring formula optimal?

(4) How well can LWPs work without proximity or cooccurrence data? If good results can be obtained without this data, computational costs would be reduced, and servers would be more likely to support the method.

If the model of distributed information retrieval underlying this article were to be adopted widely, it might be useful to formulate a standardized probe protocol for network information servers, possibly as an extension to the STARTS protocol [Gravano et al. 1997].

ACKNOWLEDGMENTS

The Center for Intelligent Information Retrieval at the University of Massachusetts at Amherst provided the script for splitting the TREC-5 data into subcollections. Ellen Voorhees of the U.S. National Institute of Standards and Technology (NIST) performed the automatic topic similarity computations using hardware and SMART software provided by Chris Buckley of SABIR Research Inc. NIST and various copyright holders granted access to the TREC data collection, TREC topics, and relevance judgments. Access to the Fujitsu AP1000 was made available under the ANU-Fujitsu Laboratories Collaborative Research Agreement. Nick Craswell contributed to discussions on the work, and Kathy Griffiths suggested improvements to the manuscript.

Sincere thanks are extended to these people and organizations and to the anonymous reviewers whose comments have resulted in significant improvements.

REFERENCES

Allan, J., Ballesteros, L., Callan, J. P., Croft, W. B., and Lu, Z. 1995. Recent experiments with INQUERY. In *Proceedings of the 4th Text Retrieval Conference* (TREC-4, Washington, D.C., Nov.), D. K. Harman, Ed. National Institute of Standards and Technology, Gaithersburg, MD, 49–63.

Buckley, C., Singhal, A., and Mitra, M. 1996. Using query zoning and correlation within SMART. In *Proceedings of the 5th Text Retrieval Conference* (TREC-5, Gaithersburg, MD, Nov.), E. M. Voorhees and D. K. Harman, Eds. National Institute of Standards and Technology, Gaithersburg, MD, 105–118.

Buckley, C., Singhal, A., Mitra, M., and Salton, G. 1995. New retrieval approaches using SMART. In *Proceedings of the 4th Text Retrieval Conference* (TREC-4, Washington, D.C., Nov.), D. K. Harman, Ed. National Institute of Standards and Technology, Gaithersburg, MD, 25–48.

Callan, J. P., Lu, Z., and Croft, W. B. 1995. Searching distributed collections with inference networks. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '95, Seattle, WA, July 9–13), E. A. Fox, P. Ingwersen, and R. Fidel, Eds. ACM Press, New York, NY, 21–28.

Chakravarthy, A. S. and Haase, K. B. 1995. NetSerf: Using semantic knowledge to find Internet information archives. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '95, Seattle, WA, July 9–13), E. A. Fox, P. Ingwersen, and R. Fidel, Eds. ACM Press, New York, NY, 4–11.

Clarke, C. L. A., Cormack, G. V., and Burkowski, F. J. 1995. Shortest substring ranking MultiText experiments for TREC-4. In *Proceedings of the 4th Text Retrieval Conference* (TREC-4, Washington, D.C., Nov.), D. K. Harman, Ed. National Institute of Standards and Technology, Gaithersburg, MD, 295–304.

Dolin, R., Agrawal, D., Dillon, L., and El Abbadi, A. 1996. Pharos: A scalable distributed architecture for locating heterogeneous information sources. Tech. Rep. TRCS96-05. Department of Computer Science, University of California at Santa Barbara, Santa Barbara, CA.

Dumais, S. T. 1992. LSI meets TREC: A status report. In *Proceedings of the 1st Text Retrieval Conference* (TREC-1, Gaithersburg, MD, Nov.), D. K. Harman, Ed. National Institute of Standards and Technology, Gaithersburg, MD, 137–152.

Fox, E. A., Ingwersen, P., and Fidel, R., Eds. 1995. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. (SIGIR '95, Seattle, WA, July 9–13). ACM Press, New York, NY.

Gravano, L. and García-Molina, H. 1996. Generalizing GlOSS to vector-space databases and broker hierarchies. In *Proceedings of the 21st VLDB Conference* (Zurich, Switzerland). VLDB Endowment, Berkeley, CA.

Gravano, L., Chang, K., García-Molina, H., Lagoze, C., and Paepcke, A. 1997. STARTS—Stanford protocol proposal for Internet retrieval and search. Computer Systems Laboratory, Stanford Univ., Stanford, CA. Available via http://www-db.stanford.edu/˜gravano/starts_home.html.

Harman, D. K., Ed. 1995. *Proceedings of the 4th Text Retrieval Conference*. (TREC-4, Washington, D.C., Nov.). National Institute of Standards and Technology, Gaithersburg, MD.

Hawking, D. and Bailey, P. 1997. Parallel document retrieval engine (PADRE) web page. Australian National University, Canberra, Australia. http://cap.anu.edu.au/cap/projects/text_retrieval.

HAWKING, D. AND THISTLEWAITE, P. 1995. Proximity operators—So near and yet so far. In *Proceedings of the 4th Text Retrieval Conference* (TREC-4, Washington, D.C., Nov.), D. K. Harman, Ed. National Institute of Standards and Technology, Gaithersburg, MD, 131–143.

HAWKING, D. AND THISTLEWAITE, P. 1996. Relevance weighting using distance between term occurrences. Tech. Rep. TR-CS-96-08. Department of Computer Science, Australian National Univ., Canberra, Australia. Available via http://cs.anu.edu.au/techreports/1996/index.html.

HAWKING, D., THISTLEWAITE, P., AND BAILEY, P. 1996. ANU/ACSys TREC-5 experiments. In *Proceedings of the 5th Text Retrieval Conference* (TREC-5, Gaithersburg, MD, Nov.), E. M. Voorhees and D. K. Harman, Eds. National Institute of Standards and Technology, Gaithersburg, MD, 359–376.

HORIE, T., ISHIHATA, H., SHIMIZU, T., AND IKESAKA, M. 1991. AP1000 architecture and performance of LU decomposition. In *Proceedings of the 1991 International Conference on Parallel Processing*. 634–635.

KIRK, T., LEVY, A. Y., SAGIV, Y., AND SRIVASTAVA, D. 1995. The information manifold. In *Papers from the AAAI Spring Symposium on Information Gathering in Distributed Heterogenous Environments* (Menlo Park, CA, Mar.), C. Knoblock and A. Levy, Eds. AAAI Press, Menlo Park, CA, 85–91.

LU, Z., CALLAN, J. P., AND CROFT. W. B., 1996. Measures in collection ranking evaluation. Tech. Rep. TR96-39. Department of Computer Science, University of Massachusetts, Amherst, MA.

MILLER, G. A. 1995. WordNet: A lexical database for English. *Commun. ACM 38*, 11 (Nov.), 39–41.

GUPTA, N. K., VOORHEES, E. M., AND JOHNSON-LAIRD, B. 1995. Learning collection fusion strategies. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '95, Seattle, WA, July 9–13), E. A. Fox, P. Ingwersen, and R. Fidel, Eds. ACM Press, New York, NY, 172–179.

VOORHEES, E. M. AND HARMAN, D. K., Eds. 1996. *Proceedings of the 5th Text Retrieval Conference*. (TREC-5, Gaithersburg, MD, Nov.). National Institute of Standards and Technology, Gaithersburg, MD.

YUWONO, B. AND LEE, D. L. 1997. Server ranking for distributed text retrieval systems on the Internet. In *Proceedings of the 5th International Conference on Database Systems for Advanced Applications* (Melbourne, Australia, Apr.), R. Topor and K. Tanaka, Eds. World Scientific Publishing Co., Inc., River Edge, NJ, 41–49.

Received: March 1997; revised: November 1997 and March 1998; accepted: April 1998