

## Singapore Management University Institutional Knowledge at Singapore Management University

---

Research Collection School Of Information Systems

School of Information Systems

---

8-1999

# Cluster-based database selection techniques for routing bibliographic queries

Jian XU


Ee Peng LIM

Singapore Management University, [eplim@smu.edu.sg](mailto:eplim@smu.edu.sg)

Wee-Keong NG

**DOI:** [https://doi.org/10.1007/3-540-48309-8\\_9](https://doi.org/10.1007/3-540-48309-8_9)

Follow this and additional works at: [https://ink.library.smu.edu.sg/sis\\_research](https://ink.library.smu.edu.sg/sis_research)

 Part of the [Databases and Information Systems Commons](#), and the [Numerical Analysis and Scientific Computing Commons](#)

---

### Citation

XU, Jian; LIM, Ee Peng; and NG, Wee-Keong. Cluster-based database selection techniques for routing bibliographic queries. (1999). *Database and Expert Systems Applications: 10th International Conference, DEXA'99 Florence, Italy, August 30 - September 3, 1999: Proceedings*. 1677, 100-109. Research Collection School Of Information Systems.

**Available at:** [https://ink.library.smu.edu.sg/sis\\_research/975](https://ink.library.smu.edu.sg/sis_research/975)

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email [libIR@smu.edu.sg](mailto:libIR@smu.edu.sg).

# Cluster-Based Database Selection Techniques for Routing Bibliographic Queries

JIAN XU      EE-PENG LIM      WEE-KEONG NG  
Centre for Advanced Information Systems (CAIS)  
School of Applied Science, Nanyang Technological University  
Nanyang Avenue, Singapore 639798  
Email: p140977971@ntu.edu.sg, aseplim@ntu.edu.sg, awkng@ntu.edu.sg

## Abstract

Given the large number of databases on the Internet, it is increasingly difficult for users to identify the databases relevant to their queries. Instead of broadcasting a given query to all databases, one would like to intelligently select only a small subset of databases for evaluating the query in order to reduce the amount of network and I/O overheads. This problem, also known as **query routing**, can be divided into three sub-problems known as **database selection**, **query evaluation**, and **result merging**. In this paper, we address the database selection problem for routing bibliographic queries. By clustering bibliographic records and summarizing their statistics, we are able to construct a knowledge base for each database and use it for database selection. We have proposed different database selection techniques based on different combinations of clustering algorithms and database ranking formulas. All these techniques have been experimented using carefully constructed bibliographic databases and their results are reported in this paper.

**KEYWORDS:** database selection, distributed databases, internet application.

## 1 Introduction

Given a user query and a set of data sources at different locations, **query routing** refers to the general problem of evaluating the query against most relevant data sources and integrating the results returned from the data sources. As Internet is now populated by a large number of web sites and databases, query routing is fast becoming one of the most important problems to be addressed on the Internet. Depending on the type of queries and the type of data sources, different forms of query routing can be defined. In the following, we illustrate some specific application examples in which query routing could be essential:

- *Scenario 1 (Global Digital Library System)*: In a global digital library system that is built upon multiple text collection servers on the Internet, a query routing problem may be defined by determining the most relevant text collection(s) for any user query which includes keyword criteria specified on the title, author and/or subject of the distributed text documents. Query routing, in this case, may involve searchable or non-searchable text collections, and the results from selected text collections may have to be combined together in order to form a single result set for the original query.

- *Scenario 2 (Electronic Shopping)*: The Internet is predicted to be a commonplace for users to perform electronic shopping. The promise of electronic shopping depends to a large extent upon the user interface and how users interact with the various electronic commerce agents on the Internet. Typically, each retailer will provide online information about his/her products. To ensure that a buyer with a specific buying need will be able to locate the right retailer(s) quickly, we need a query routing mechanism that can suggest fairly accurately a small number of retailers for the buyer to consider or patronize. Unlike scenario 1, the data sources to be dealt with are product information from the retail stores, and the buying need may be represented by conditions specified on the product attributes.

In [10, 24], the overall query routing problem has been divided into three inter-related sub-problems, namely *database selection*, *query evaluation* and *result merging*. The three sub-problems also represent the three sequential steps to be performed in query routing.

Database selection refers to the problem of analysing a user query and determining one or more appropriate data sources at which information satisfying the user query can be located. In order to address the database selection problem, *essential* knowledge about the content of individual data sources has to be acquired. Query evaluation refers to the problem of dispatching and evaluating the user query to the data sources chosen in the database selection step. Due to possible heterogeneous data representations at different data sources, the original user query may have to be translated into different local query statements to be submitted to the remote data sources. Result merging refers to the problem of integrating results returned by different data sources. Apart from standardizing the local result formats, one may have to re-compute the rank information of the integrated query result because of the different ranking formulas adopted by the data sources.

## 1.1 Objective and Scope

In this paper, we focus on the database selection problem in the context of a global digital library consisting of a large number of online bibliographic servers. Each server hosts a bibliographic database that contains bibliographic records each of which consists of text values for a number of pre-defined bibliographic attributes such as *title*, *author*, *call number*, *subject*, etc.. Each bibliographic database supports user queries on the bibliographic attributes. Since bibliographic records are relatively small in size, we only consider bibliographic databases that support boolean queries on the bibliographic attributes, and the query results are not ranked. While focusing on bibliographic databases, we believe that our proposed solution can be extended to handle other databases that contain searchable text attributes.

Formally, we define a general **database selection problem** as follows:

**Definition 1** *Let  $D$  be a set of databases,  $q$  be a query, and  $M$  be the number of databases which should be selected (or query  $q$  should be forwarded to). Compute  $E \subseteq D$  such that  $|E| = M$  and  $(\forall F \subseteq D \text{ such that } |F| = M, \text{ Goodness}(q, E) \geq \text{Goodness}(q, F))$ . ■*

In the above definition, *Goodness* measures the degree of relevance for the combined result returned by the selected data sources. Gravano and Garcia-Molina, in [7, 8, 9], proposed a few possible *Goodness* functions that can be adopted by database selection. In routing bibliographic queries, we have adopted a *Goodness* function defined below:

**Definition 2** Given a set of bibliographic databases  $\{db_1, \dots, db_N\}$  denoted by  $E$  and a query  $q$ ,

$$Goodness(q, E) = \sum_{i \in E} s_i$$

where  $s_i$  denotes the result size returned by  $db_i$  for query  $q$ . ■

Several database selection solution techniques have been developed for a collection of full text databases, e.g. NCSTRL<sup>1</sup> and TREC<sup>2</sup> collections. Very few techniques, on the other hand, have been proposed for bibliographic databases which contain multiple text-based attributes[9, 24].

The crux of the database selection problem is to construct a knowledge base that captures the content of local data sources well enough that the degree of relevance of each data source with respect to any given query can be determined accurately. Usually, the knowledge base maintains some statistical information that characterizes the content of each data source. In [9] and [24], term frequencies from each bibliographic server have been used to construct the knowledge base. Nevertheless, in a large database, database records can often be classified into different groups such that each group of records share some common or similar values for some attributes. Using clustering techniques, we would like to discover such grouping of database records. By capturing some essential statistics about each cluster of records, we hope to further improve the accuracy of database selection techniques.

In this paper, we investigate the use of clustering to improve the accuracy of database selection. Several cluster-based database selection techniques have been proposed to route bibliographic queries. Unlike other non clustered-based approaches, cluster-based database selection techniques involve clustering of database tuples before the content of each database is summarized. We have proposed different database selection techniques by coupling three clustering techniques known as *Single Pass Clustering* (SPC), *Reallocation Clustering* (RC), and *Constrained Clustering* (CC) with two database ranking formulas namely Estimated Result Size (ERS) and Estimated Goodness Score (EGS). To evaluate the performance of cluster-based database selection techniques, experiments have been conducted systematically using collections of bibliographic databases specially constructed to demonstrate skewness in their content.

## 1.2 Paper Outline

The rest of this paper is structured as follows. Section 2 provides a brief survey of the relevant work in database selection. In Section 3, we give an overall description of the cluster-based database selection techniques. Section 4 describes three database clustering techniques. Following that, two cluster-based database ranking formulas known as ERS and EGS are given in Section 5. The performance evaluation experiments of database selection techniques built upon combination of database clustering techniques and database ranking formulas are reported in Section 6. Finally, we conclude the paper and describe our future work in Section 7.

---

<sup>1</sup>NCSTRL (Networked Computer Science Technical Reference Library) [15] consists of distributed online collections of technical reports that can be queried through web interface.

<sup>2</sup>TREC (Text REtrieval Conference) [22] consists of specially selected collections of text documents provided by NIST (National Institute of Standards and Technology). These collections are designed to be used by researchers to conduct information retrieval experiments and to compare their results.

## 2 Related Work

In recent years, different forms of database selection problems have been studied by several research groups, and various solution approaches have been proposed. As Gravano pointed out in [10], database selection problems can occur both in *routing* and *mediating* queries to distributed data sources. Query routing is often carried out for a set of text collections or collections with text attributes such that the collections share a common and simple schema. Query mediation, on the other hand, involves heterogeneous schemas exported by the underlying databases and the schemas usually complement one another in the database content. While query routing often adopts a query model which returns partial results to any given queries, query mediation requires complete query results to be returned from the participating databases.

In the following, we describe previously proposed approaches to select databases for query routing. These research solutions can be classified into three main categories depending on the type of databases to be handled.

### 2.1 Database Selection for Text Collections

Research efforts in this category deal with collections of text documents. Usually, the *vector space* retrieval model is adopted for querying the text collections. A query supported by such a model consists of a set of keywords, and the relevance of a text document is determined by the frequency of keywords appearing in the document and their discriminatory power.

In the gGLOSS project[7, 9], the document frequencies of terms found in every text collection are computed and included in the knowledge base for database selection. Using the document frequencies, the relevance of each text collection can be estimated for a given user query.

In Callan's work, the CORI (*Collection Retrieval Inference Network*) project[2], the  $TF \times IDF$  document ranking method has been extended to rank a set of text collections where  $TF$  denotes *term frequency* and  $IDF$  denotes *inverse document frequency*. In this method, the  $TF \times IDF$  document scoring formula is modified by replacing  $TF$  and  $IDF$  by  $DF$  and  $ICF$  (*inverse collection frequency*) respectively. A CORI network is later constructed based on the relationship between collections and their terms, and the relationship between a given query and its term. Each collection is scored using the CORI network and is determined by the combined belief or probability of all query terms. It is assumed that all terms involved in the query are of equal importance.

Based on the document frequency knowledge, Yuwono and Lee proposed a unique database ranking formula based on Cue-Validity Variance (CVV)[25]. The proposed database ranking formula essentially incorporates the discriminatory power of keywords across collections. It was shown that the CVV-based database selection technique out-performed the database selection techniques in gGLOSS and CORI.

### 2.2 Database Selection for Collections with Multiple Attributes

In the GLOSS (*Glossary of Servers Server*) project[8, 9, 20], a database selection technique for collections containing multiple text attributes has been proposed. Different from gGLOSS, the *boolean* retrieval model is adopted for querying collections in GLOSS. The queries for such collections consist of keyword predicates on the different attributes such as *author*, *title*, etc. Given a collection and an attribute-term pair, the number of records having the attribute values containing the term is known as the *frequency of the attribute-term pair*. This frequency information has been further used to estimate the rank of each database. The main assumption behind GLOSS is that terms appearing

in any specific attribute of records of a collection follow independent and uniform distributions. The discriminatory power of each term is not considered in this work. Real user queries and a set of six databases (INSPEC, COMPENDEX, ABI, GEOREF, ERIC and PSYCINFO databases) have been used to evaluate the performance of GLOSS.

### 2.3 Database Selection for Collections Accessible Through Query Interface Only

So far, the database selection techniques given in Sections 2.1 and 2.2 assume that the document frequency information for each text collection is available for query routing. This is possible either by having full access to the text collections or by mandating each text collection to provide the necessary information voluntarily. Nevertheless, in reality, not all text collections may be able to cooperate fully on providing their local information. Hence, one may have to investigate database selection techniques for collections accessible through query interface only.

Voorhees[21] proposed two database selection techniques for text collections with vector space query interface. In the two techniques, known as *multiple relevant document distribution* (MRDD) and *query clustering* (QC), text collections are ranked based on their responses to the training queries most similar to the query to be routed. Although these methods do not require a large knowledge base and are easy to implement, it is not clear how training queries that sufficiently capture the content of a database can be generated. Furthermore, the two techniques only deal with text collections.

In our recent research[24], we have designed new database selection techniques for distributed bibliographic databases using training queries and their query results. It has been shown that using statistical information compiled from query results, it is possible to perform database selection with reasonable accuracy.

## 3 Overview of Cluster-based Database Selection

Clustering refers to the grouping of database records based on the degrees of similarity between the records. Clustering has been used in many fields, such as information retrieval (IR)[19, 4, 14], data mining, data reduction[1], etc. In order to route queries to a set of databases each with multiple text attributes, the content of each databases has to be summarized properly. Nevertheless, as the databases contain wide range of information, direct summarization of their content may result in inaccurate summary knowledge. In such cases, clustering may be applied to discover the hidden grouping of database records. By summarizing the content of different groups of database records, we believe that the accuracy of summary knowledge can be improved. Example 1 that follows will illustrate how clustering would improve database selection.

**Example 1** *A text collection consisting of 20 documents has a group of 5 documents related to IR (information retrieval) and another group of 15 documents related to Databases. Below is a table showing the frequencies of two selected terms that appear in the collection.*

<i>Term</i>	<i>Term frequency with respect to the entire collection (20 documents)</i>	<i>Term frequency with respect to IR documents (5 documents)</i>	<i>Term frequency with respect to Database documents (15 documents)</i>
index	10	5	5
inverted	5	4	1

If only the term frequencies with respect to the entire collection are used to estimate the number of documents containing both `index` and `inverted`, an estimated number of documents that contain both terms can be computed by using the simple probability theory<sup>3</sup> as:  $20 \times \frac{10}{20} \times \frac{5}{20} = 2.5$ . This estimation is not accurate because it assumes that all terms follow independent and uniform distributions. In this example, this assumption is not true since `index` and `inverted` are highly correlated among the IR related documents but not the Databases related documents. Hence, if we can successfully identify these two groups of documents (i.e., IR group and Databases group) by some clustering technique, it is estimated that  $5 \times \frac{5}{5} \times \frac{4}{5} = 4$  documents from IR group and  $15 \times \frac{5}{15} \times \frac{1}{15} = \frac{1}{3}$  from Database group contain both terms. The latter estimation is significantly different from the former one without considering the documents grouping (i.e., clustering). ■

In this paper, we therefore propose a few cluster-based database selection techniques and apply them to the query routing problem over a set of bibliographic databases. To apply the clustering techniques, the issues below have to be addressed:

- What are the clustering algorithms? How are the similarity between two bibliographic records, and similarity between a bibliographic record and a cluster defined?
- How is a cluster represented? What statistical knowledge has to be captured for each cluster for database selection purpose?
- How many clusters should be generated for each bibliographic database? How does the number of clusters affect the database selection performance?

Our proposed cluster-based database selection techniques involve two steps namely *knowledge construction* and *database ranking* as shown in Figure 1. In knowledge construction, bibliographic records from each database involved are clustered and the content of each cluster is summarized. Database ranking is then performed for a given query based on the summary information of the clusters from each database. The query will be matched against the clusters from each database and a matching score will be computed for each cluster. When a query matches well with a cluster, it is likely that many records in that cluster will be relevant to the query. Furthermore, when a query matches well with significant number of clusters from a database, it is likely that the database will be more relevant to the query. In this case, the rank of each database will be determined by the matching scores between its clusters and the given query as well as the cluster sizes.

## 4 Clustering Techniques for Bibliographic Databases

Clustering of text documents is a well researched problem in information retrieval[3, 16, 12, 13]. Nevertheless, to our best knowledge, there has not been much work in clustering bibliographic databases consisting of multiple text attributes. We therefore have adapted some text clustering techniques to cluster bibliographic databases.

### 4.1 Similarity Between a Bibliographic Record and a Cluster

Clustering can only be performed on the bibliographic databases when the similarity between a bibliographic record and a cluster can be determined and quantified. In this section, the similarity measures used in our proposed clustering techniques are defined.

---

<sup>3</sup>The estimation is based on the GLOSS[8, 9] technique.

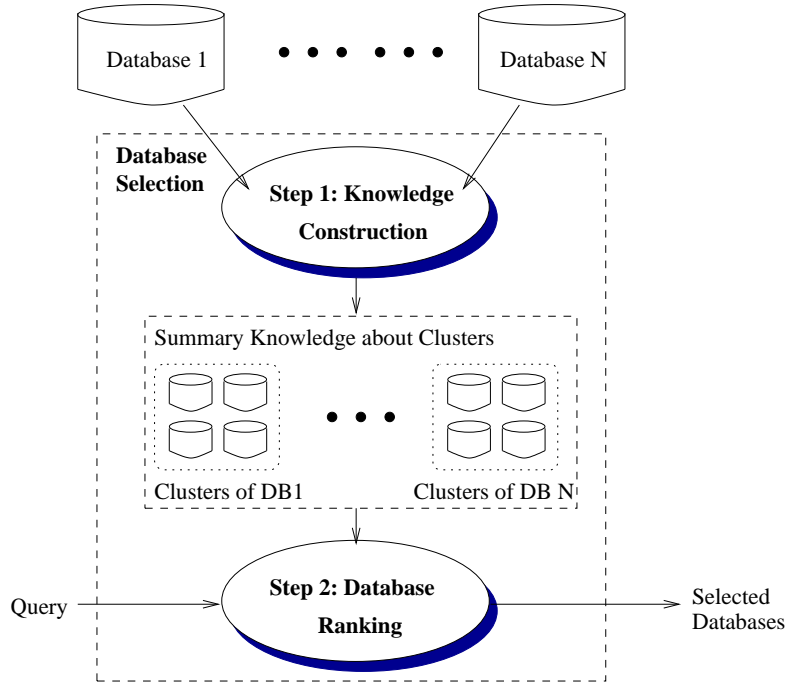


Figure 1: Cluster-based database selection steps

Several attributes can be found in a bibliographic database, e.g. *title*, *subject*, etc. In this paper, we only deal with text attributes and assume that all databases contain the same set of attributes<sup>4</sup>. An attribute value can be described by an attribute descriptor (to be defined later) which is essentially a text vector. A database record can be represented by a set of attribute descriptors, one for each attribute. Similarly, a cluster can also be represented by a set of attribute descriptors. Hence, the similarity between a bibliographic record and a cluster can be defined based on the attribute descriptor information representing the record and the cluster.

**Definition 3** Let  $t_1, t_2, \dots, t_W$  be all possible terms in our term dictionary, an **attribute descriptor** is defined by a vector  $v = (w_1, w_2, \dots, w_W)$ , where  $w_j$  denotes the term weight of term  $t_j$ . ■

Apart from being used to represent bibliographic records, attribute descriptors can also be used to represent clusters and queries. In the case of bibliographic records, term frequencies are used as term weights (denoted by  $w_j$ 's in the above definition).

**Definition 4** Let  $A_1, A_2, \dots, A_l$  be bibliographic attributes, a **bibliographic record**  $r$  is defined by a vector of attribute descriptors, one for each attribute,

$$r = (vr_1, vr_2, \dots, vr_l).$$

---

<sup>4</sup>If different attribute sets are found in different bibliographic databases, a uniformed attribute set still can be adopted by integrating the different attribute sets.



Since each term will be counted at most once for each attribute with respect a record,  $vr_k$ 's are binary vectors. A term weight of 1 will be assigned when term  $t_j$  appears in the record for the respective attribute. Otherwise, a term weight of 0 will be assigned.

**Definition 5** A cluster consisting of a set of bibliographic records,  $r_1, r_2, \dots, r_{N_c}$ , is defined by a binary tuple

$$c = (N_c, D_c),$$

where  $D_c$  is a list of attribute descriptors

$$D_c = (vc_1, vc_2, \dots, vc_l),$$

and

$$vc_k = vr_{1,k} + vr_{2,k} + \dots + vr_{N_c,k}$$

where  $vr_{i,k}$  ( $0 \leq i \leq N_c$ ) denotes the  $k$ th attribute descriptor of  $r_i$ . ■

In the above definition,  $D_c$  captures the representative content of all bibliographic records belonging to a cluster.

**Definition 6** The similarity between a bibliographic record  $r (= (vr_1, \dots, vr_l))$  and a cluster  $c (= (N_c, (vc_1, \dots, vc_l)))$ , denoted by  $SIM_{r,c}$ , is defined as:

$$SIM_{r,c} = \frac{1}{l} \sum_{k=1}^l SIM_{vr_k,vc_k} \quad (1)$$

where the  $SIM_{vr_k,vc_k}$  denotes the **similarity between the bibliographic record and the cluster with respect to attribute  $A_k$** , and is defined by the cosine distance<sup>5</sup> between the two vectors:

$$SIM_{vr_k,vc_k} = \frac{|vr_k \cdot vc_k|}{|vr_k| \cdot |vc_k|} \quad (2)$$

■

As shown in Formula (1), the similarity between a cluster and a bibliographic record is defined by averaging the similarities between the record and the cluster for all bibliographic attributes.

**Example 2** Consider a bibliographic database that contains two attributes namely  $A_1 = \text{title}$ ,  $A_2 = \text{subject}$ , and a term dictionary of  $W=3$ . Let the terms be **information**, **retrieval**, and **clustering** assigned with term ids 1,2 and 3 respectively. The two bibliographic records below can be represented by  $r_1 = ((1, 0, 1), (0, 0, 1))$  and  $r_2 = ((1, 1, 0), (1, 1, 0))$ , respectively.

record ids	title	subject
1	information clustering	clustering
2	information retrieval	information retrieval

Let  $c_1 = (10, ((2, 2, 3), (2, 1, 5)))$  be a cluster containing 10 bibliographic records. The two attribute descriptors  $(2, 2, 3)$  and  $(2, 1, 5)$  contain the term weights for the title and subject attributes, respectively. Figure 2 shows the bibliographic records  $r_1$ ,  $r_2$  and cluster  $c_1$  in a three-dimension space.

Using Formula (1), the similarity between the bibliographic record  $r_1$  and cluster  $c_1$  can be computed as follows:

---

<sup>5</sup>The Cosine coefficient[4] is originally proposed to calculate the similarity between two document vectors. We have borrowed the formula for measuring the similarity between a bibliographic record and a cluster.

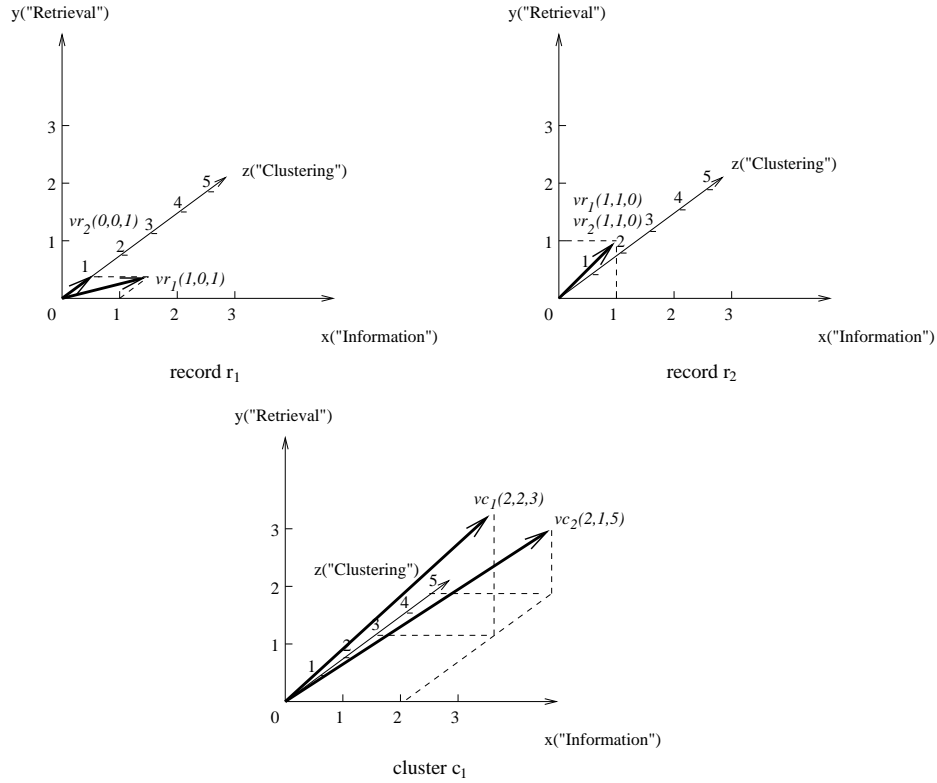


Figure 2: Representation of Records and Clusters

$$\begin{aligned}
 SIM_{r_1, c_1} &= \frac{1}{2} \cdot \left( \frac{(1 \times 2 + 0 \times 2 + 1 \times 3)}{\sqrt{(1^2 + 0^2 + 1^2) \cdot (2^2 + 2^2 + 3^2)}} + \frac{(0 \times 2 + 0 \times 1 + 1 \times 5)}{\sqrt{(0^2 + 0^2 + 1^2) \cdot (2^2 + 1^2 + 5^2)}} \right) \\
 &= 0.885
 \end{aligned}$$

The similarity between the bibliographic record  $r_2$  and cluster  $c_1$  is:

$$\begin{aligned}
 SIM_{r_2, c_1} &= \frac{1}{2} \cdot \left( \frac{(1 \times 2 + 1 \times 2 + 0 \times 3)}{\sqrt{(1^2 + 1^2 + 0^2) \cdot (2^2 + 2^2 + 3^2)}} + \frac{(1 \times 2 + 1 \times 1 + 0 \times 5)}{\sqrt{(1^2 + 1^2 + 0^2) \cdot (2^2 + 1^2 + 5^2)}} \right) \\
 &= 0.537 \quad \blacksquare
 \end{aligned}$$

## 4.2 Proposed Database Clustering Techniques

Single Pass Clustering (SPC) and Reallocation Clustering (RC) are two straightforward clustering techniques used for text documents[4]. To cluster bibliographic databases, the two techniques have been modified to cater for bibliographic records consisting of multiple text attributes. In addition, we have proposed a Constrained Clustering (CC) technique that generates for a bibliographic database a fixed number of clusters specified by the user. These three clustering techniques have been used with two different database ranking formulas given in Section 5.

### 4.2.1 Single Pass Clustering Technique (SPC)

Single pass clustering technique is basically a greedy algorithm that always assigns a bibliographic record to the most similar cluster. Since each bibliographic record is read only once, SPC technique is efficient and easy to implement. Nevertheless, SPC technique requires a similarity threshold  $TH$

specified by the user.  $TH$  is used to determine if the similarity between a record and a cluster is high enough to assign the record to the cluster. When  $TH$  is small, each cluster can accommodate records that are less similar. Hence, a smaller number of clusters will be generated. The detailed clustering steps are given below:

1. For each bibliographic record from the database, perform Steps (2) and (3).
2. Find the most similar cluster for the record among the existing clusters using the similarity measure between a bibliographic record and a cluster (see Formula (1)).
3. If no cluster has been created so far, or the similarity measures between the record and all existing clusters are lower than the given threshold  $TH$ , a new cluster containing the record is created. Otherwise, the record will be inserted into the cluster that is most similar.
4. All outlier clusters (clusters containing only 1 or 2 records) are combined into one.

Although the single pass clustering technique is simple, it is criticized[4] for its tendency to produce large clusters early in the clustering process. This situation also appears in our experiment using the single pass clustering technique. It is because that the clusters generated by the SPC technique depends on the order in which bibliographic records are processed.

#### 4.2.2 Reallocation Clustering (RC)

Reallocation clustering[4, 6] operates by selecting an initial set of clusters followed by some iterations of re-assigning bibliographic records to the most similar clusters. Through the iterations, the cohesiveness among records in a cluster is improved. The following algorithm describes the steps required by the reallocation clustering technique for a bibliographic database.

1. Apply SPC to the database and use the clusters generated by SPC as the initial clusters. These include the outlier cluster.
2. For each bibliographic record from the database, perform Steps (3) and (4).
3. Find the most similar cluster for the record among the given clusters (see Formula (1)).
4. If the similarity measures between the record and all given clusters are smaller than the given threshold  $TH$ , the record will be inserted into the outlier cluster. Otherwise, the record will be inserted into the cluster which is most similar.
5. After all records have been re-assigned, re-calculate the cluster vectors.
6. The resultant clusters of Step (5) are used as the input set of clusters for the next iteration of reallocation (i.e., Step (2) is performed again) until a specified number of iterations are completed (or no bibliographic record is assigned to different cluster in an entire iteration).

In reallocation clustering, it is difficult to decide how many iterations should be executed. For simplicity, we have chosen 9 iterations in our experiments as described in Section 6. Like SPC, RC relies on a user specified threshold to indirectly control the number of clusters generated.

### 4.2.3 Constrained Clustering (CC)

For both SPC and RC, there is no control parameter that directly controls the storage requirement for the generated cluster information. The number of resultant clusters is controlled indirectly by the threshold  $TH$ . To overcome this shortcoming, we proposed the **Constrained Clustering (CC)** technique. CC is able to generate a fixed number  $\beta$  of clusters for each database where  $\beta$  is specified by the user. Like in the case of RC, CC requires an initial set of clusters to be first generated followed by iteratively improving the similarity among records within the clusters. The algorithm is given below:

1. Use the first  $\beta$  largest clusters generated by SPC as the initial clusters where clusters which contain most records are called the largest clusters.
2. For each bibliographic record from the database, insert it into the most similar cluster.
3. After all records have been processed, recalculate the cluster vectors.
4. The resultant clusters of Step (3) are used as the input set of clusters for the next iteration of reallocation (i.e., Step (2) is performed again) until a specified number of iterations are completed.

## 4.3 Discussions

Several relevant issues regarding to our proposed database clustering techniques for bibliographic databases are discussed as follows.

### 4.3.1 Outliers

Outliers[6] are records that are dissimilar to almost all other records. It is difficult to fit them into even the most similar cluster, i.e., the distance from the bibliographic record to its most similar cluster is much larger than the distance between any pair of records in that cluster. In this case, we have to decide whether outliers should be included into the most similar clusters (despite that they may not be similar enough) or to generate new clusters for them.

If we allow outliers to be included into individual clusters consisting of only one or two outlier records, large amount of storage resources will be required. On the other hand, if outliers are forced to be included into some clusters containing other records, the accuracy of clustering will be compromised. This becomes a trade-off between the clustering accuracy and the storage requirement of knowledge base. In SPC and RC clustering techniques for bibliographic database, all outliers are combined into a single cluster known as the outlier cluster. In CC clustering technique, outlier cluster is not required. The reason is that there are not specific cluster(s) designated for outlier records in CC technique and the number of clusters is determined prior to clustering.

### 4.3.2 Insignificant Terms

Since the size of term dictionary is usually very large and the term frequency distribution is governed by the Zipf's Law, some clustering techniques [17, 18] eliminate those *insignificant terms (IST)* which have very small term frequencies. These insignificant terms are eliminated on the basis that they have insufficient discriminatory power for objects to be clustered. We have also conducted experiments to evaluate the performance of clustering using IST elimination (shown in Section 6).

## 5 Cluster-Based Database Ranking Formulas

In this section, two cluster-based database ranking formulas are given. They are defined based on the similarity between a given query and a database represented by a set of clusters.

**Definition 7** A query is defined to be a list of attribute descriptors, i.e.

$$q = (vq_1, vq_2, \dots, vq_l),$$

where  $vq_k$  is an attribute descriptor with respect to attribute  $A_k$ . ■

Each attribute descriptor  $vq_k$  of a query captures the search terms specified for attribute  $A_k$ . (Since a search term will only appear at most once for each attribute in the query,  $vq_k$ 's are binary vectors.) A term weight of 1 will be assigned when term  $t_j$  is given in the query for the respective attribute. Otherwise, a term weight of 0 will be assigned.

**Example 3** Consider the bibliographic database in Example 2, a query consisting of the following predicate: `subject = (information and clustering)` can be represented by  $q = (\vec{0}, (1, 0, 1))$ .  $\vec{0}$  denotes a zero text vector for the title attribute while  $(1, 0, 1)$  denotes the text vector for the subject attribute. ■

Once a set of clusters have been generated for each database, we can apply the following two database ranking formulas to compute the rank of the databases for a given query.

### 5.1 Cluster-based Database Ranking based on Estimated Result Sizes (ERS)

This database ranking scheme computes the database rank by estimating the query result size returned by a database. The estimated result size returned, originally proposed in GLOSS[8, 9], can be computed by summing the estimated query result sizes returned by clusters belonging to the database.

**Definition 8** The estimated result size (ERS) of a given query  $q$  from database  $db_i$  is defined as:

$$\mathcal{E}_{db_i, q} = \widehat{Size}_{(db_i, q)} = \sum_{n=1}^{|C|} \widehat{Size}_{(c_n, q)} \quad (3)$$

where  $C = \{c_1, c_2, \dots, c_{\beta_i}\}$  is a set of clusters generated for database  $db_i$ , and  $\widehat{Size}_{(c_n, q)}$ , the estimated result size of a given query  $q$  returned from a cluster  $c_n$ , is defined as:

$$\widehat{Size}_{(c_n, q)} = |c_n| \cdot \prod_{\substack{k=1 \\ vq_k \neq \vec{0}}}^{|A|} \prod_{\substack{j=1 \\ w'_{j,k} \neq 0}}^W \frac{w_{j,k,n}}{|c_n|} \quad (4)$$

where  $w'_{j,k}$  denotes the weight of the  $j$ th term in the attribute descriptor  $vq_k$  for query  $q$ ,  $w_{j,k,n}$  denotes the term frequency of the term  $t_j$  in the attribute descriptor  $vc_k$  of cluster  $c_n$ , and  $|c_n|$  denotes the number of records in the cluster  $c_n$ . ■

In the above definition, we assume that all attributes in a cluster are independently distributed and all terms in an attribute domain are also independently distributed. The predicates  $vq_k \neq \vec{0}$  and  $w'_{j,k} \neq 0$  indicate that only terms appearing in the query  $q$  and their corresponding terms appearing in cluster  $c_n$  will be considered in the computation. Note that Formula 3 and 4 can be seen as an extension to the goodness function adopted by GLOSS[9]. When  $\beta_i = 1$ , i.e., there is only 1 cluster for  $db_i$ , our proposed ranking formula reduces to that of GLOSS.

## 5.2 Cluster-based Database Ranking based on Estimated Goodness Score (EGS)

Instead of estimating the query result size returned from each database, the EGS ranking formula, extending that adopted by Yuwono and Lee[25], computes the goodness score of a database with respect to a given query by using CVV to rank databases with multiple attributes.

**Definition 9** *The estimated goodness score (EGS) of database  $db_i$  for a given query  $q$  is defined as follows:*

$$\mathcal{E}_{db_i,q} = \sum_{n=1}^{|C|} \mathcal{E}_{c_n,q} \quad (5)$$

where  $C = \{c_1, c_2, \dots, c_{\beta_i}\}$  is a set of clusters generated for database  $db_i$ .

The estimated goodness score  $\mathcal{E}_{c_n,q}$  of cluster  $c_n$  with respect to query  $q$  is defined by:

$$\mathcal{E}_{c_n,q} = \prod_{\substack{k=1 \\ vq_k \neq \vec{0}}}^{|A|} \sum_{\substack{j=1 \\ w'_{j,k} \neq 0}}^W CVV_{j,k} \cdot w_{j,k,n} \quad (6)$$

where  $CVV_{j,k}$  denotes the variance of  $CV_{i,j,k}$ 's, the Cue Validity of term  $t_j$ , for attribute  $A_k$  across all databases,  $w'_{j,k}$  denotes the weight of term  $t_j$  in attribute  $A_k$  for query  $q$ ,  $w_{j,k,n}$  denotes the term frequency of term  $t_j$  with respect to attribute  $A_k$  in cluster  $c_n$ . ■

## 6 Experiments

To evaluate the performance of database selection techniques that are built upon various combination of the three database clustering techniques and the two cluster-based database ranking formulas, a number of experiments have been conducted.

We conducted the experiments using the same experiment framework adopted in our previous work[24]. The difference is that we do not use training queries but the summary knowledge about clusters. The experiments have been designed to answer a few questions about cluster-based database selection techniques:

- Does the clustering algorithm used affect the performance of database selection techniques?
- How does a cluster-based database selection technique perform when different number of clusters are generated?
- How does a cluster-based database selection technique perform for bibliographic databases with different skewness in their content?

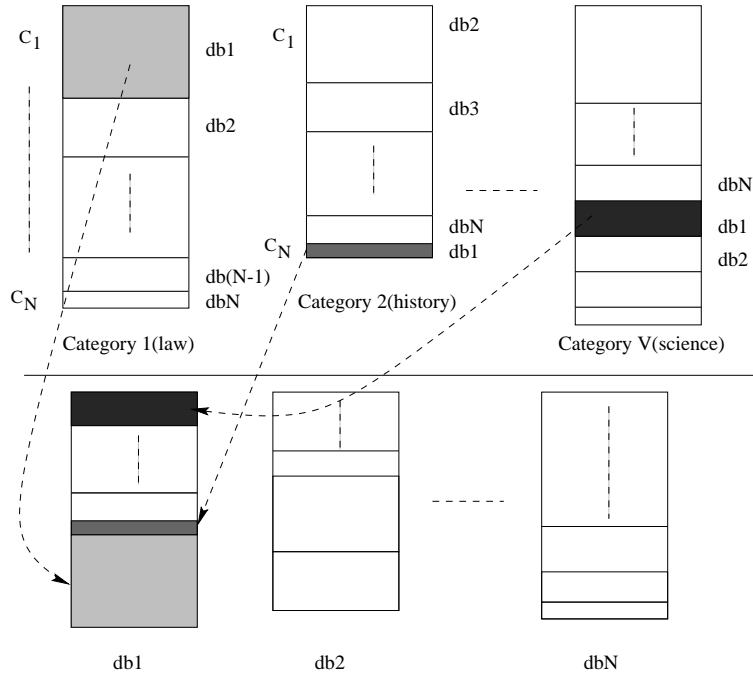


Figure 3: Organize the catalogue records from categories to databases

- How much storage requirement do our cluster-based database selection techniques need compared to non cluster-based database selection techniques?

In the rest of this section, we describe the experiment setup, and the performance measure used. The experimental findings of our cluster-based database selection techniques are presented and analyzed.

## 6.1 Experiment Framework

To set up the bibliographic database collection for our experiments, we down-loaded all bibliographic records from NTU<sup>6</sup> library database. NTU library database contains 217,928 bibliographic records. The records are classified according to the Library of Congress (LC) classification scheme. For example, the call number QA76.9.D3.AI49 indicates that the bibliographic record belongs to the *mathematics science* category.

For our experiments, a collection of 10 smaller bibliographic databases has been constructed ( $N = 10$ ) using the down-loaded bibliographic records based on the following strategy:

- All bibliographic records are grouped according to their LC categories. Assume that there are  $V$  such virtual categories and we want to assign their records to  $N$  databases such that each database contains records from all categories, and at the same time contains distinct makeup of records from different categories. In this way, the databases in our collection always demonstrate different degrees of relevance for the same query.
- We divide each category into  $N$  groups, with records assigned to the groups according to the following pre-defined ratio (the sizes of these groups are determined by the Zipf-like

<sup>6</sup>The web page of Nanyang Technological University library is available at: (<http://web.ntu.ac.sg/library/>).

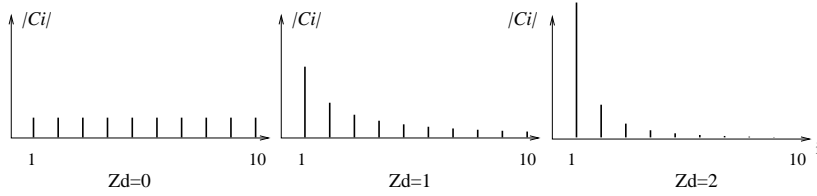


Figure 4: Category distribution given different database skew value (when  $N=10$ )

distribution[5, 11, 23]):

$$|C_1| : |C_2| : \dots : |C_N|$$

where:

$$|C_i| = \frac{|C|}{i^{Z_d} \sum_{j=1}^N \frac{1}{j^{Z_d}}} \quad (7)$$

$|C|$  is the size of the category,  $Z_d$  is **Database Skew**. (When  $Z_d > 0$ ,  $|C_i|$  has a Zipf-like distribution, and when  $Z_d = 0$ , it is a uniform distribution. )

- The groups are assigned to  $N$  databases in a round-robin manner.

The assignment of bibliographic records to different databases in our experiment is illustrated by Figure 3. Three LC categories (i.e., *law*, *history* and *science* categories) are shown in this figure. For instance, database  $db_1$  is constructed (indicated by the arrows) by records from these three categories with different ratio (which are shown in the figure with different pattern).

By varying the  $Z_d$  value, we can evaluate the performance of database selection techniques in database collections with different skewness. When  $Z_d = 0$ , each category is evenly distributed to the  $N$  databases. It should be noted that the larger  $Z_d$  is, the more skew is each category being grouped[11] (see Figure 4). In particular,  $Z_d = 1$  was selected as a normal database skew level so that we can evaluate the performance of our techniques when a static database skew is required. On the other hand, the different degrees of database skew,  $Z_d = 0, 0.5, 1, 1.5,$  and  $2$  were used in our experiments to evaluate the performance of our techniques for database collection with different database skews.

## 6.2 Performance Measurement

Since it is difficult or even impossible to find the ideal clusters for databases, we did not attempt to evaluate the performance of database clustering techniques. We only focus on the performance of our entire cluster-based database selection techniques.

In our experiments, a performance measure (denoted by  $P$ ) derives the accuracy of a database selection technique by computing the ratio between the combined result size returned by the database selection technique and that returned by the ideal choice of databases. Given  $K$  test queries  $\{q_1, q_2, \dots, q_K\}$ ,  $P$  is computed as follows:

$$P = \frac{1}{K} * \sum_{j=1}^K P_j \quad (8)$$



where  $P_j(1 \leq j \leq K)$  represents the performance contributed by test query  $q_j$ .

$$P_j = \frac{\sum_{db_i \in G} s_{i,j}}{\sum_{db_i \in B} s_{i,j}} \quad (9)$$

$G$  represents the set of databases selected by a proposed database selection technique. The ideal database selection is  $B$ .  $s_{i,j}$  denotes the actual result size of test query  $q_j$  returned by database  $db_i$ . Clearly  $0 \leq P \leq 1$ . When  $M = N$ ,  $G = B$  and  $P = 1^7$ .

Furthermore, we use the same set of 2000 synthetic test queries, which have been adopted in [24], to evaluate the performance of these techniques. The synthetic test queries are generated as follows.

- *Step 1:* Randomly select a record from the combined set of bibliographic records collected from all experimental databases.
- *Step 2:* Extract title and subject values from the record.
- *Step 3:* Randomly decide whether to use title, subject or both in a new query.
- *Step 4:* For each attribute (title or subject) to be included in the query, construct a predicate on it by randomly selecting one to four distinct terms from the corresponding extracted attribute value. No stop words are used in this step.

Moreover, the minimum result size for the test queries is fixed to 2.

Using the performance metric and synthetic test queries, we can make comparison between different database selection techniques.

### 6.3 Parameter Setting

The experiments are conducted by varying or fixing the following parameters which are used to perform the cluster-based database selection:

- $M$  - the number of databases to be selected ( $M = 1, 2, \dots, 10$ )
- $\beta$  - the number of clusters (only available for CC method, three values were selected  $\beta = 20, 50, 100$ )
- $TH$  - the threshold of the similarity between a record and a cluster to decide whether to combine the record into the cluster (Five values were selected  $TH = 0, 0.05, 0.1, 0.2, 0.4$ . It is available for SPC and RC methods while it only decides the initial clusters for CC method)
- $Lp$  - the number of iterations (fixed to 9, for RC and CC methods)
- $O$  - the minimum number of records in a normal cluster (fixed to 3 in this experiment. It implies that all clusters containing only two or one records will be considered as outlier clusters and be combined into one cluster)
- $Z_d$  - the database skew ( $Z_d = 0, 0.5, 1, 1.5, 2$ )

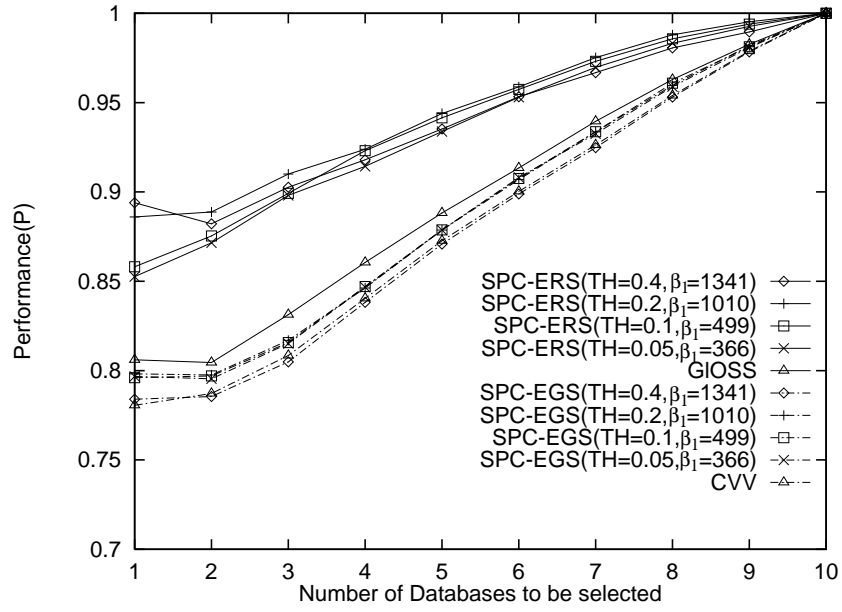


Figure 5: Performance of Database Selection Techniques using SPC ( $Z_d=1$ )

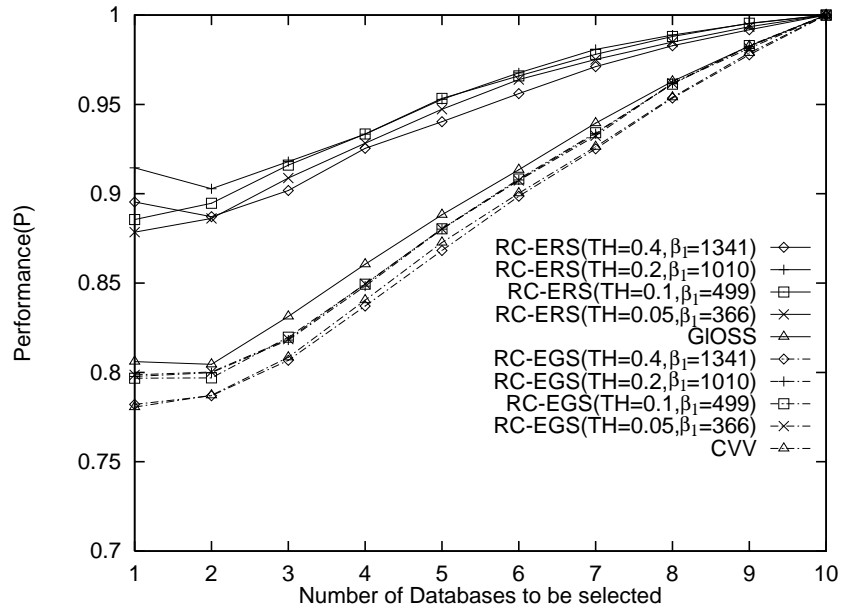


Figure 6: Performance of Database Selection Techniques using RC ( $Z_d=1$ )

## 6.4 Experimental Findings

Figures 5 to 9 show the performance  $P$  of different database selection techniques by varying the number of databases to be selected ( $M$ ). For techniques using SPC and RC clustering algorithms, the similarity threshold  $TH$  is a parameter that indirectly controls the number of clusters generated for each database. In Figures 5 and 6, we show the performance of techniques using SPC and RC when different  $TH$  values are adopted<sup>8</sup>. For techniques using CC clustering algorithm, the number of clusters in each database is directly controlled by the parameter  $\beta$ , where the  $TH$  is only used in generating the initial set of clusters using SPC (e.g.,  $TH = 0.2$  and  $\beta = 50$  for CC mean that CC clustering uses the first 50 largest clusters generated by SPC with  $TH = 0.2$  as the initial clusters). Figures 7 to 9 show the performance of technique using CC for the number of clusters  $\beta = 20, 50$  and 100, respectively. Different initial clusters decided by  $TH$  are also adopted in these figures. In order to be compared with CC, the numbers of clusters generated<sup>9</sup> for database  $db_1$  by SPC and RC are shown in Figures 5 and 6.

To compare the performance of our proposed database selection techniques with that of others and our previous database selection techniques, the performance of GLOSS[8, 9] and CVV[25] are also shown in each of these figures as the baselines. Note that GLOSS database selection technique could be considered as the extreme case for our cluster-based database selection technique using only one cluster (i.e.,  $\beta = 1$ ) and ERS as the database ranking formula. On the other hand, CVV technique could be considered as the extreme case of the cluster-based database selection technique using one cluster and EGS as the database ranking formula. Since all our proposed techniques outperform random database selection significantly, we do not show the performance of random database selection in these figures.

From the experiments conducted, we have several findings as described below:

- Cluster-based database selection techniques using ERS significantly outperform those using EGS. The exact database clustering technique used does not even affect the performance of database selection techniques using EGS. This case occurs especially when a larger number of clusters were generated. When the number of clusters increases, database selection techniques using ERS outperforms those using EGS. In [25], the GLOSS database selection technique is shown to perform better than the database selection technique using CVV for a set of text documents. In our experiments, we notice that the same phenomenon also occurred in the case of databases containing multiple text attributes.
- All database selection techniques using SPC and RC with similarity threshold  $TH = 0.2$  usually outperform those using different similarity threshold. When a similarity threshold higher than 0.2 is chosen, the condition to combine records into clusters becomes stringent and the number of clusters increases. Moreover, the number of outliers will also increase. By combining the outliers into a outlier cluster (see Section 4.2), a large number of records will be stored in the outlier cluster. This in turns reduces the accuracy of clustering technique and worsens the performance of our proposed database selection techniques. On the other hand, clustering techniques using similarity thresholds lower than 0.2 will generate a small number

---

<sup>7</sup> $M = |G| = |B|$ .  $M$  is the number of databases to be selected.

<sup>8</sup>In these figures, SPC-ERS denotes the database selection technique with ERS database ranking formula using SPC as the clustering method. This convention of naming database selection techniques will be used henceforth.

<sup>9</sup>As each database may have different number of clusters generated by SPC or RC, we only show the number of clusters for database  $db_1$  (denoted by  $\beta_1$ ). Note that SPC and RC could generate slightly different numbers of clusters due to the possibility that some of the clusters may not be assigned any record during the reallocation phase in RC.

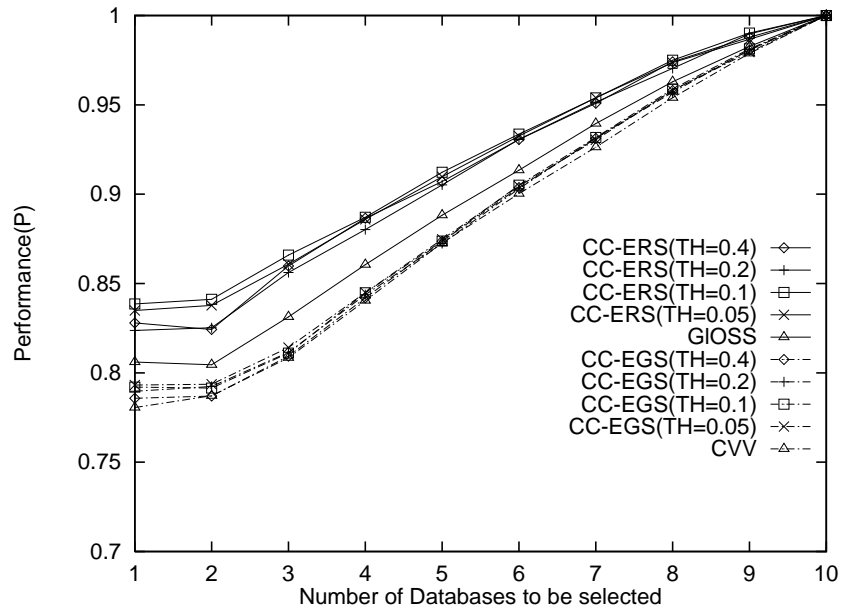


Figure 7: Performance of Database Selection Techniques using CC with  $\beta = 20$  ( $Z_d=1$ )

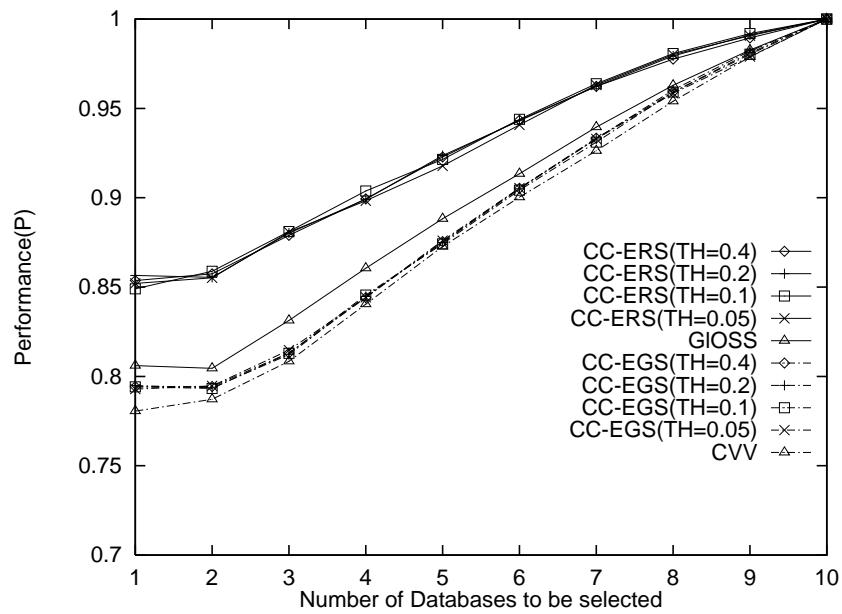


Figure 8: Performance of Database Selection Techniques using CC with  $\beta = 50$  ( $Z_d=1$ )

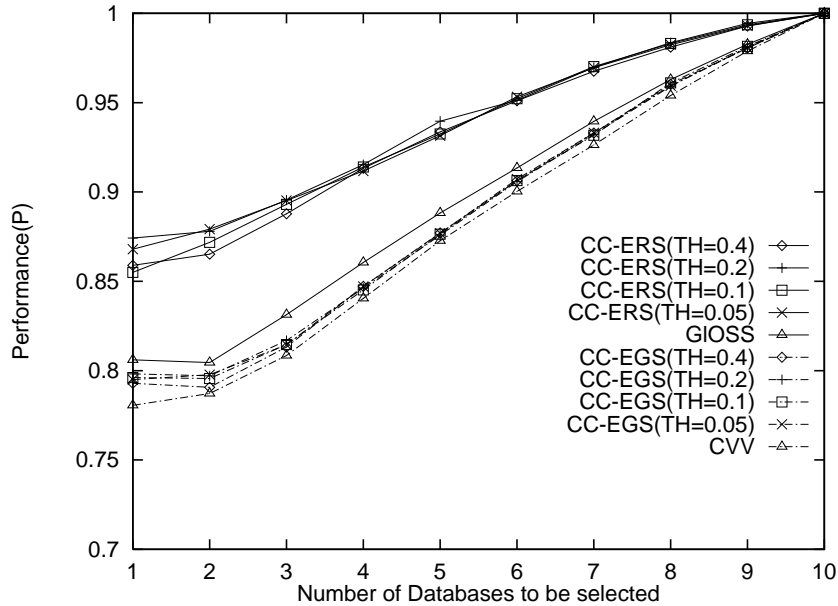


Figure 9: Performance of Database Selection Techniques using CC with  $\beta = 100$  ( $Z_d=1$ )

of clusters for each database. This might not reflect the exact distribution of database and will also compromise the database selection performance. In our experiments, it was shown that techniques using  $TH = 0.2$  yield relatively good performance.

- Database selection techniques using RC perform slightly better than those using SPC. Re-allocation method reassigns all records into clusters based on the actual distribution of the database after enough times of iterations. The clustering becomes more accurate after several times of iterations using RC than using SPC which only processes each record once.
- For CC technique, the choice of initial set of clusters is not important. For a given number of clusters,  $\beta$ , no matter what similarity threshold ( $TH$ ) was chosen, the performance of our database selection techniques using CC is similar (see Figures 7 to 9).

The performance of database selection using CC clustering technique is shown in Figure 10. In the figure, different  $\beta$ 's have been adopted while fixing  $TH = 0.2$ .

- As shown in Figure 10, for CC, the larger is the number of clusters, the more accurate is the performance of the database selection techniques.

To evaluate the performance of database selection of eliminating the insignificant terms, we have conducted the experiment that applies the elimination of IST in the CC clustering technique (this procedure is denoted by ECC in the figure). The performance of this technique is shown in Figure 11. We found that:

- Insignificant Terms (IST) elimination is not useful for database selection.

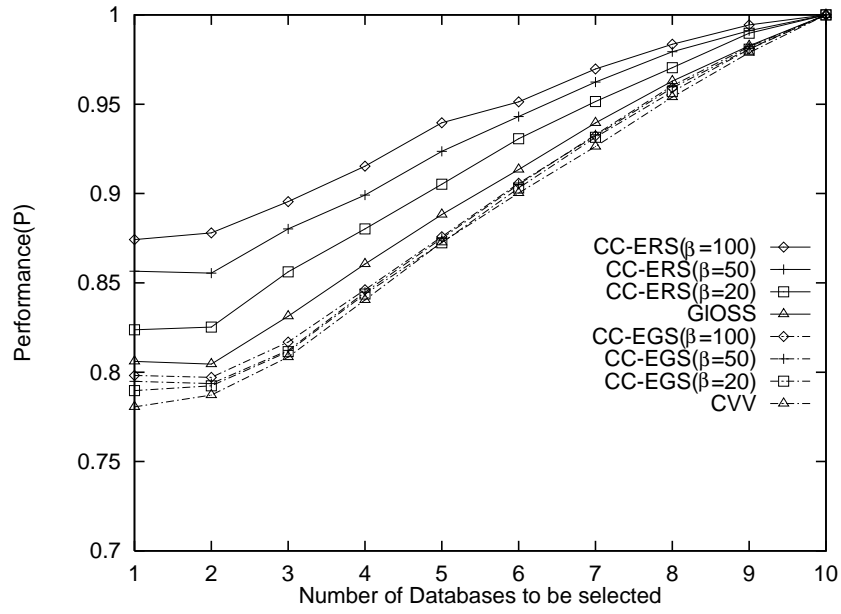


Figure 10: Performance of Database Selection Techniques using CC when different number of clusters is fixed, ( $TH=0.2, Z_d=1$ )

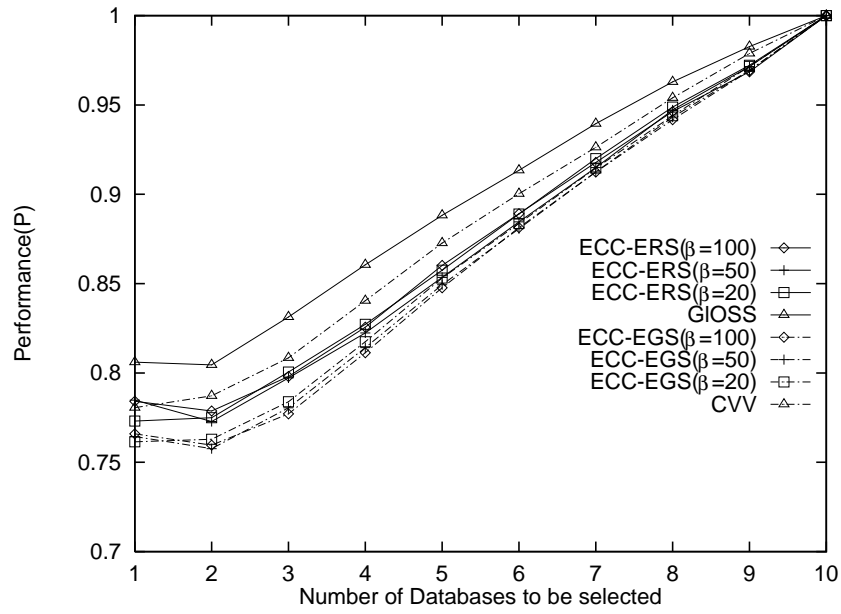


Figure 11: Performance of Database Selection Techniques using CC considering IST (Insignificant Terms) elimination( $TH=0.2, Z_d=1$ )

$TH$	Storage Space needed for Database Selection Techniques				overhead (hours)
	0.05	0.1	0.2	0.4	
RC-*	7372712	7617025	7849960	6574284	30
SPC-*	7047737	7347421	7573306	6204695	0.8
CC-*( $\beta=100$ )	6575474	6610165	6669754	6684034	5
CC-*( $\beta=50$ )	5889197	5918678	5924371	5939144	3
CC-*( $\beta=20$ )	4805016	4804310	4815553	4801659	2
GLOSS/CVV	3349915				0.5
ECC-*( $\beta=100$ )	2003444				
ECC-*( $\beta=50$ )	1978756				
ECC-*( $\beta=20$ )	1841404				

Table 1: The storage space needed (bytes) and average computing overhead (hours) for Database Selection Techniques with different similarity threshold ( $Z_d=1$ )

Furthermore, we investigate the storage requirement and computing overhead of database selection techniques using SPC, RC, CC (with  $\beta=20, 50, 100$ ), together with that of GLOSS and CVV as the baselines<sup>10</sup>. Table 1 shows the results.

- *RC* clustering technique has the largest storage requirement. The storage needed by *CC* can be adjusted by  $\beta$  and is highly lower than *SPC* and *RC*. On the other hand, the computing overhead of *CC* is also highly lower than that of *RC* as the number of clusters is directly controlled. We further find that *CC* (Constrained Clustering) needs relatively lower storage requirement as well as computing overhead and has acceptable performance.

The performance of our cluster-based techniques in different types of database skew values are shown in Figure 12 as well. The results are promising. Our proposed database selection techniques RC-ERS, SPC-ERS and CC-ERS always outperform GLOSS and CVV techniques in all database skew values. Even when the database skew is 0 (i.e, the databases are randomly distributed), our proposed database selection techniques still have good performance. In particular, RC-ERS outperforms GLOSS by 12% when  $Z_d = 0$  while it only has about 5% improvement when  $Z_d = 2$ . SPC-ERS and CC-ERS also have promising performance.

## 7 Conclusions

Query routing is a common class of problems that involve selecting the appropriate information sources for a query to be evaluated, and merging the query results from the selected sources. In this paper, we have proposed several cluster-based database selection techniques (SPC, RC and CC clustering, and ERS, EGS database ranking formulas). Unlike our previous database selection research that were proposed to route bibliographic queries using training queries, these database selection techniques are derived by combining three database clustering techniques with two database ranking formulas. Through experiments, we have shown that cluster-based database selection techniques outperform non cluster-based database selection techniques. However, clustering techniques require storage space more than their non cluster-based counterparts. In cases where accuracy of database selection outweighs the storage overheads, cluster-based database selection techniques could be applied.

As part of our future work, we plan to pursue the following research directions:

<sup>10</sup>The storage requirement of ECC are also shown as reference.

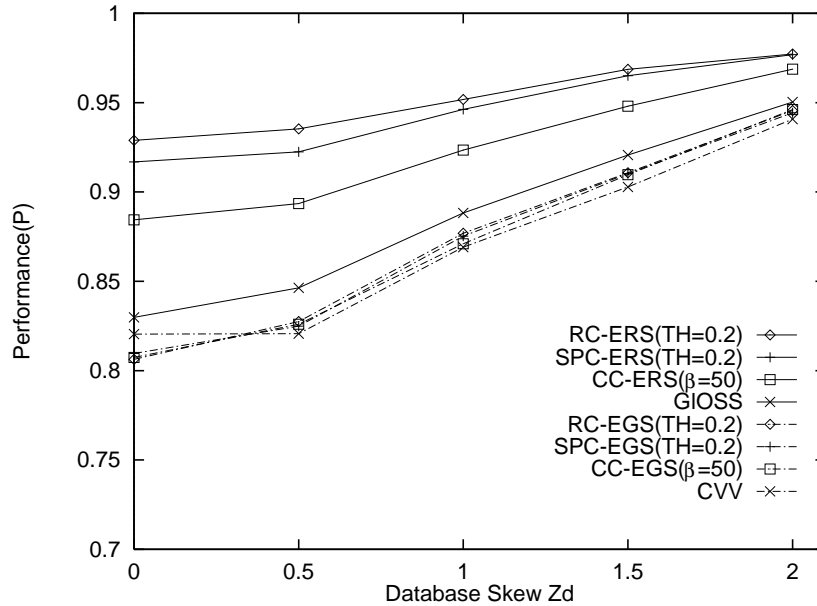


Figure 12: Performance of different cluster-based Database Selection Techniques as a function of database skew value  $Z_d$  ( $M=5$ ,  $TH=0.2$  for SPC and RC,  $\beta = 50$  for CC)

- *Implementation:* We believe that these proposed database selection techniques can be applied to select bibliographic servers on the Internet. Hence, we plan to develop a query routing broker that incorporates the suitable database selection techniques for a distributed technical report collection. We will investigate the system issues involved in building such an intelligent broker and address them accordingly.
- *Database evolution:* Due to the time constraint, we have not investigated the database evolution issue. The three database selection techniques have to be extended to update their knowledge base as the databases evolve in their content. It is also important to keep the overhead of updating the knowledge bases low so that database selection can still be efficiently performed.
- *Improvement on Query Attribute Modeling:* Currently, our proposed techniques treat several bibliographic attributes (i.e., author, title, subject) uniformly. We assume that these attributes are independently distributed. However, these assumption do not correspond to reality. We plan to look into placing different importance to different attributes in our future experiment.

## References

- [1] D. Barbara, W. DuMouchel, C. Faloutsos, P.J. Haas, J.M. Hellerstein, Y. Ioannidis, H.V. Jagadish, T. Johnson, R. Ng., V. Poosala, K.A. Ross, and K.C. Sevcik. The new jersey data reduction report. *Bulletin of the Technical Committee on Data Engineering*, 20(4), December 1997.



- [2] J.P. Callan, Z. Lu, and W.B. Croft. Searching Distributed Collections With Inference Networks. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28, 1995.
- [3] D.R. Cutting, D.R. Karger, J.O. Pederson, and J.W. Tukey. Scatter/Gather: a cluster-based approach to browsing large document collection. In *Proceedings of ACM/SIGIR*, pages 318–329, 1992.
- [4] W.B. Frakes and R. Baeza-Yates. *Information Retrieval : data structures & algorithms*. Englewood Cliffs, N.J., 1992.
- [5] M.A. Golberg. *An Introduction to Probability Theory with Statistical Applications*. Plenum Press. New York and London, 1984.
- [6] M. Goldszmidt and M. Sahami. A Probabilistic Approach to Full-Text Document Clustering. Technical Report ITAD-433-MS-98-044, SRI International, 1998.
- [7] L. Gravano and H. Garcia-Molina. Generalizing GLOSS to Vector-Space Databases and Broker Hierarchies. In *Proceedings of the 21st International Conference on Very Large Data Bases (VLDB'95)*, pages 78–89, Zurich, Switzerland, September 1995.
- [8] L. Gravano, H. Garcia-Molina, and A. Tomasic. The Effectiveness of GLOSS for the Text Database Discovery Problem. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 126–137, Minneapolis, Minnesota, May 1994.
- [9] L. Gravano, H. Garcia-Molina, and A. Tomasic. GLOSS: Text-Source Discovery over the Internet. *ACM Transactions on Database Systems (To Appear)*, 24(2), June 1999.
- [10] L. Gravano and Y. Papakonstantinou. Mediating and Metasearching on the Internet. *Bulletin of the Technical Committee on Data Engineering*, 21(2), June 1998.
- [11] K.A. Hua, Y-L. Lo, and H.C. Young. Considering Data Skew Factor in Multi-Way Join Query Optimization for Parallel Execution. *VLDB Journal*, 2(3):303–330, July 1993.
- [12] D. Koller and M. Sahami. Hierarchically Classifying Documents Using Very Few Words. In *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*, pages 170–178, San Francisco, CA, 1997.
- [13] S.H. Li and P.B. Danzig. Boolean Similarity Measures for Resource Discovery. *IEEE Transactions on Knowledge and Data Engineering*, 9(6), November/December 1997.
- [14] Charles T. Meadow. *Text Information Retrieval Systems*. San Diego : Academic Press, 1992.
- [15] NCSTRL. <http://www.ncstrl.org>.
- [16] M. Sahami, M. Hearst, and E. Saund. Applying the Multiple Cause Mixture Model to Text Categorization. In *Proceedings of the 13th International Conference on Machine Learning (ICML'96)*, pages 435–443, San Francisco, CA, 1996.
- [17] M. Sahami, S. Yusufali, and M.Q.W. Baldonado. Real-time Full-text Clustering of Networked Documents. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI'97)*, page 845, Menlo Park, CA, 1997.

- [18] M. Sahami, S. Yusufali, and M.Q.W. Baldonado. SONIA: A Service for Organizing Networked Information Autonomously. In *Proceedings of the 3rd ACM International Conference on Digital Libraries (DL'98)*, Pittsburgh, Pennsylvania, USA, June 1998.
- [19] G. Salton. *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Publishing Company, 1988.
- [20] A. Tomasic, L. Gravano, C. Lue, P. Schwarz, and L. Haas. Data Structures for Efficient Broker Implementation. *ACM Transactions on Information Systems*, 15(3), July 1997.
- [21] G. Towell, E.M. Voorhees, N.K. Gupta, and B. Johnson-Laird. Learning Collection Fusion Strategies for Information Retrieval. In *Proceedings of the 12th Annual Machine Learning Conference*, Lake Tahoe, July 1995.
- [22] Text REtrieval Conference (TREC). <http://trec.nist.gov>.
- [23] C.L. Viles and J.C. French. Dissemination of Collection Wide Information in a Distributed Information Retrieval System. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 12–20, 1995.
- [24] J. Xu, Y.Y. Cao, E.P. Lim, and W.K. Ng. Database Selection Techniques for Routing Bibliographic Queries. In *Proceedings of the 3rd ACM International Conference on Digital Libraries (DL'98)*, Pittsburgh, Pennsylvania, USA, June 1998.
- [25] B. Yuwono and D.L. Lee. Server Ranking for Distributed Text Retrieval Systems on the Internet. In *Proceedings of the 5th International Conference on Database Systems for Advanced Applications (DASFAA '97)*, pages 41–49, Melbourne, Australia, April 1997.