Resource Discovery in Distributed Digital Libraries

Norbert Fuhr
University of Dortmund, Germany

Abstract

In the near future, users will have access to a vast number of digital libraries. For a given information need and limited resources, there is the problem of selecting those libraries which produce an overall optimum answer. This resource discovery problem is additionally complicated by the diversity of the sources, e.g. with respect to media, document formats, indexing methods, database schemas and protocols. Once a set of digital libraries has been selected, the collection fusion problem deals with the problem of merging the answers of these libraries in order to get a high retrieval quality. This paper describes the specific problems and gives an overview on the solutions that have been developed so far.

1 Introduction

Resource discovery in digital libraries deals with information search in a networked, heterogeneous environment where a large number of digital libraries are accessible. To the user, this environment should make the impression of a single large virtual library. This way, it would be possible to exploit all the accessible knowledge in order to satisfy the information needs of a user.

At first glance, the World Wide Web seems to provide already a solution to this problem: By putting the content of all digital libraries as documents on the Web, the virtual digital library would be reality. However, besides the copyright and commercial issues that stand against such a solution, the major drawback would be the insufficient support of the semantics of the digital documents — ranging from external attributes to complex internal representations of documents. There is slow progress towards solving some of these problems for the Web in general — e.g. by the Dublin Core ([Weibel 95]) set of standard attributes of Web documents, XML ([Connolly 97]) as a means for modeling semantic document structure and RDF ([Miller 98]) as an effort for supporting interoperability of Web-based information services in general. Although these efforts may solve some of the problems of resource discovery, it is obvious that digital libraries have specific needs that can not be fulfilled by general Web-based information services and protocols.

In this paper, we will first outline the major dimensions of the problem space and then address systems and content-related issues in more detail[1]. For the latter, we will focus on two specific problems of distributed information retrieval, namely database selection and collection fusion. Finally, we will give an outlook on future work.

2 Problem space and research issues

The problems associated with resource discovery in digital libraries differ substantially from those in traditional information retrieval. While some of the differences are issues of scope — orders of magnitude increase in the number of resources — others are due to technical and organization contexts that are not a factor in traditional information discovery. These include the interaction of the following factors:

- The non-deterministic behavior that results from the complex interactions in widely distributed systems. The availability and response time of components of a digital library implemented as a distributed system may vary due the complex interaction of unrelated or loosely-related factors. These include of network load, server load, network partitioning due to external factors, and human behavior.
- Diversity in the nature of information. This diversity may exist along four dimensions that play a role in the way people search for, access, and use information — media (such as text, audio, video, etc.), format (such as the encoding of image files in GIF, JPEG, TIFF, etc.), logical structure (such as books, diaries, anthologies), and semantics.
- Diversity in the types of users of the information system. Not only are there many types of users, but the spread between high-end and low-end users is getting larger.
- Diversity in the types of devices and network connectivity that provide access to information. Again, the spread between high-end and low-end devices and connectivity is getting larger.

[1] This part of the paper is based on the whitepaper "Resource Discovery in a Globally-Distributed Digital Library" produced by the joint NSF-EU Digital Libraries Working Group on resource indexing and discovery in 1998.

- Increase in the number of protocols for locating information. These include proprietary database protocols and open standards such as Z39.50.

- The local administration and creation of information, protocols, and services. While coordination among these local parties is often a goal, the reality is that distributed digital libraries are federations combining the interests of numerous stake holders, often with conflicting interests.

- A diversity of notions of what is the "best" response to a query. Complex factors such as information quality, information overload, timeliness, and economics often play vital roles in determining the appropriate query response.

- The commodity nature of information and the attendant incentives to abuse the "information marketplace" with various forms of misinformation and attempts to distort system behaviors.

- Diversity in the type of discovery-related services required by this broader spectrum of users, varying from straightforward ad hoc retrieval to complex filtering, summarization, and the creation of dynamically updated information spaces.

The interaction of these factors can be summarized by some rather broad statements about the solution space. The solution to distributed information access will not be created by imposing a single monolithic solution on everybody. In addition, the best technical solutions may be impossible to achieve due to non-technical factors such as political or legal constraints. Instead, the solution space for distributed searching must be expressed in terms of multiple targets, acceptable behaviors, and layered solutions. Given a set of acceptable levels of service and functionality for the variety of organizational, economic, and user contexts, what are the layered solutions varying from no cooperation, to loose agreements, to tightly coupled organizations?

In order to solve these problems, there is a need for research in the following areas:

Organization: Given a diverse set of institutions, the task is to develop methods of cooperation in order to provide access to their information in a manner that is coherent and effective for users.

Systems: Ensure acceptable behavior in the face of limited and varied computation resources and connectivity and decentralized control.

Content: Handle large quantities of content and the unlimited variety of content forms and types.

Human computer interface: On the input/output level, the task is to get useful inputs from the users to display results in a meaningful way. On the cognitive level, the system is trying to understand what the user is trying to do and helping the user to understand what the system is actually doing.

Research facilities, metrics and measurements: For the evaluation of resource discovery methods, test beds, taxonomies and metrics have to be developed.

In the remainder of this paper, we will focus on the systems and content issues.

3 Systems issues

Access to distributed information is hampered by the multiplicity of material available on-line from a network of public, private and commercial organizations, libraries, publishers, vendors and individuals. There is a great need for the development of a system infrastructure that facilitates navigation and retrieval, and that provides mediating support for the maze and variety of information available on-line. This system infrastructure should be capable of identifying, accessing, and retrieving the digital resources available. Furthermore, it needs to provide a coherent and consistent view of as many of the information repositories as possible.

The overall goal of system architecture is to ensure acceptable behavior in the face of limited and varied computation resources and connectivity and decentralized control. The goal of system architecture is also to attempt future proofing where possible and permit scalability as the system grows. Several properties of distributed, federated information systems complicate service guarantees:

- The variety of hardware (processors, network, protocol, display devices) complicates system optimization.

- Collection size and diversity, and the number of collections affects system behavior.

- Diverse requirements for privacy and security affect performance.

- Different models of system cooperation imply a spectrum of solutions for any given problem.

Three research threads are of central importance to the development of an architectural infrastructure that supports access to distributed information while ensuring acceptable behavior: database selection, database interaction, and consistency management.

3.1 Database Selection

Database selection at the systems level involves query routing to physical servers. This is distinguished from database selection at the content level, described in section 5, which involves logical query routing. This distinction is comparable to the difference between URLs, which are physical locations in the Internet, and URNs, which are logical names for entities that may exist at one or more physical locations.

In an environment where information is distributed across multiple repositories, system designers must develop the infrastructure that selects carefully the repositories to which a query is sent. Existing technology typically uses the trivial approach of sending the query to all repositories. This approach does not pay attention to scalability as networked bandwidth is perceived as a given commodity and because scalability requires a "logical" (top-down) design and coordinated deployment. In a domain where there are potentially one million repositories (an accurate description of today's Web) broadcast techniques are prohibitively expensive. Increasing the network bandwidth may help alleviate some of these problems, however the rapidly growing population of networked information will continue to outstrip network and database server capacity.

Network-adaptive caching, replication and application naming are used traditionally to conserve network and server resources and reduce response times. For example, it is possible to cache a subset of "hot" pages at many stages of a network hierarchy and deploy caching and replication servers at strategic locations across the Internet. Caching and replication, however, have a down side as they introduce the need to maintain updates and preserve consistency between caches/replicas and the original information. The use of wide area distributed test beds, e.g., caching and replication servers, in conjunction with good measurements of the locality of references would generate reference streams that can be used to predict the expected performance. The dependencies of performance indicators such as bandwidth consumption, and latency with parameters such as the degree of replication, document popularity, actual cache hit rates, error rates needs and the usefulness of logs to determine the locality of references need to be investigated.

Caching and replication technologies also face a number of legal constraints when deployed in distributed information environments. Information providers and rights holders may be sensitive to the degree to which caching and replication expose them to illegal copying or distribution of information. While it is widely recognized that reliability depends on these technologies, there may be constraints on their use such as decreased "time to live" on cache items or other barriers to turning temporary copies into persistent objects.

The role of metadata, which is the focus of another working group, in query routing should be investigated. This metadata could supply information about the characteristics of the server that are relevant to database selection. For example, if performance were an important criteria for query routing (get the answer as quickly as possible), routers could rely on historical load information made available by servers. Another example might be routing based on freshness of information in a set of replicated servers. Using metadata about the last update of a particular set of servers, a query router could determine the best target for a query.

3.2 Database Interaction

Once a set of candidate repositories is selected, the system infrastructure must interact with the repositories in the network. This requires bringing together repositories of heterogeneous types that are used by different organizations. The heterogeneity of the information available in these databases — different naming conventions, data structures, search engines, vocabularies for access — coupled with the wide variation in granularity and level of detail of the resources described challenge the ability to identify and retrieve needed information. The goal here is to achieve independence from data formats, document models, and languages. It is important to lower the barriers to allow access to heterogeneous materials and to provide cross-collection search capabilities. Access to existing heterogeneous resources should be provided without any relocation, reformatting and restructuring of data. Each repository should be able to use its own way to represent documents, but the documents should be exchanged freely and displayed on heterogeneous computing platforms. Integrated access could be provided to a variety of search engines, relying on different document models and query languages, by means of front-end systems responsible for query translation and mapping as well as post-result filtering.

This infrastructure should support a broad range of interaction types, inter-repository protocols, distributed search protocols (including the ability to search across heterogeneous databases with a level of syntactic/semantic consistency), and object interchange protocols. Multiple protocols may exist for multiple target points — an inter-repository protocol that is optimal for one form of federation may be inappropriate for another form. Better mechanisms for privacy, security and protection, cooperative authentication, and charging also need to be addressed.

3.3 Consistency Management

Finally, to ensure acceptable quality of service, the system infrastructure must manage several levels of consistency as documents, collections, and requirements change. Documents have a transient nature and their contents are likely to change often over time. An increasing number of documents are constructed from a dynamic pool of components to match user- and application-specific requirements. The number of short-duration, "live" applications, like collaborative white-board sessions, is increasing.

To accommodate change, existing Internet-wide search services periodically poll for changes in documents. As the number and frequency of changes increase, the amount and frequency of polling increase, consuming ever more resources. There will never be sufficient computational and network resources to ensure complete consistency between the search service index and the available documents. Thus, some flexible compromises must be reached.

Ironically, a degree of inconsistency is acceptable in certain information systems. Even though the newspaper that is delivered every morning contains information that is hours old, it is still deemed to be a valuable resource. However, a stock ticker with prices that are hours old is unacceptable for online trading.

Complex linkages between resources in distributed digital libraries increase the complexity of consistency management. Not only is there the question of maintaining consistency between exact replicates, but attention must also be paid to methods for maintaining consistency among versions, manifestations, summaries, translations, editions, and other derivative and related forms of intellectual content.

To ensure acceptable behavior, distributed resource discovery applications achieve the required degree of consistency between index and collection within the boundaries imposed by the availability of network and computational resources. Traditional solutions for consistency in distributed systems like call-backs, provide a mechanism for identifying changes in tightly coupled federated systems. More recent developments in communication based on publish/subscribe models push change events from information providers to information consumers. In general, improvements in consistency management require investigation in communication and negotiation protocols, registration and subscription services, and object description frameworks.

4 Content

Distributed resource discovery is complicated by quantity of content and the unlimited variety of content forms and types. This content may include transient objects, dynamic objects, distributed objects, and objects (works) that exist in multiple manifestations on multiple servers. This section describes content-related research issues in three subsections — content-based database selection, query language translation and mapping, and semantic heterogeneity.

4.1 Content-based Database Selection

For a given query, limited system resources prohibit forwarding the query to all possible databases. Therefore, the system has to select the databases most appropriate for answering the query. The primary goal of the selection is the optimization of retrieval quality for which size and scope of the database as well as the quality of the underlying retrieval system have to be considered. Note that this content-based selection problem is distinguished from the systems-based issues described in section 3.1 in that selection is based on content characteristics of a database (as a logical entity), rather than on systems characteristics of physical servers such as connectivity or load.

For this purpose, metadata about the databases is required, which can be of different granularity, from frequency distributions of attribute values to high-level, condensed descriptions. Other factors affecting the selection are search capabilities (e.g., for doing a geographic search), and pricing conditions. In order to perform an optimum selection, appropriate methods for deriving metadata of different granularity — for all kinds of media and representation languages — and for estimating the relevant parameters have to be developed.

A distributed environment presents distinct challenges to the creation of content metadata for query routing. Content in individual databases has idiosyncratic characteristics. Descriptive metadata about this content may vary in quality, follow different standards, or may be non-existent. Objects may be transient or "live", in the sense that their content is time-dependent.

A number of existing strategies have been developed to gather content metadata about databases. However, these methods are almost entirely text-based, calling for the development of new classes of algorithms for non-text and complex or compound objects. So far, two major strategies for database selection have been proposed, which we call synthetic vs. holistic. Synthetic strategies consider each query condition separately, regarding each database with respect to this condition, and then synthesizing the information from all query conditions. Holistic strategies consider the query as a whole, by looking for similar queries from the past and how they performed on the different databases.

One interesting case is ratings metadata. In order to guide users to the appropriate sources for their information need or for selecting the most suitable answers, ratings of databases or individual documents will be essential for future digital libraries. Major criteria for ratings will be the quality of the material, the appropriateness for the current information need and filtering with respect to the user group (e.g., children). Today, these ratings are part of a publisher's work. In digital libraries, where everybody can act as publisher, other institutions and new mechanisms will be necessary for performing the rating and for making sure that these ratings are considered when accessing digital material.

There is a need for implementing standards for the general structure and format of ratings such that a system can consider them during retrieval. For assigning the ratings, appropriate infrastructures (e.g., rating agencies) have to be established.

4.2 Query Languages

Future digital libraries will contain a variety of multimedia and hypermedia documents. For any type of document, there are three different views, namely the logical view (dealing with logical structure), the layout view (dealing with the presentation of the document) and the content view. Furthermore, documents or parts thereof may be assigned attributes. Thus, there are four different aspects that may be addressed in a query, but there is a lack of query languages integrating all these aspects (and for different media). Also, expressiveness of the query language is an important issue, since many current document retrieval languages are still based on propositional logic, whereas database languages are more expressive, but lack good mechanisms for dealing with the intrinsic uncertainty of information retrieval. Future query languages also should include operators for specifying the logical structure, layout and content of the result. In order to support interoperability, standards for such query languages have to be devised.

A crucial issue is the representation of the content view, which may be at different abstraction levels, namely syntactic (or signal-based), semantic, and pragmatic (use-oriented). For example, a photo may be described in terms of colors, textures and contours, by giving the objects displayed and their spatial distribution, or by describing the impression that it makes on a typical viewer. For non-textual media, there is a lack of automatic indexing methods generating higher-level representations. Also appropriate methods for generating these representations for multimedia documents by combining evidence from the different media components should be investigated.

Query languages may differ in the set of operators and predicates as well as their general expressiveness. Appropriate strategies for dealing with these differences are required, e.g., preprocessing of the query (possibly generating several queries for a single original query), post-processing of the results or accepting the increased imprecision.

4.3 Semantic heterogeneity

A single database may contain a variety of document types, and different databases may be based on different schemas and use different query languages. From a users' point of view, many of these differences are not relevant for their information needs. Thus, the system has to provide mechanisms for coping with semantic heterogeneity. For mapping between different schemas, ontologies may be used. When the domains of related attributes in different schemas are totally different, additional knowledge sources are required. Terminological resources support mapping between different content representations (e.g., text-, classification- or thesaurus-based) and cross-lingual retrieval requires multilingual dictionaries. Methods for automatically constructing these resources and for using them in query mapping have to be developed.

As general strategies for coping with semantic heterogeneity, automatic as well as semi-automatic methods may be devised. Automatic methods will be based on the principle of retrieval as uncertain inference, accepting (to a limited extent) imprecise mappings. Semiautomatic methods will identify semantic ambiguities and ask the user to resolve them.

5 A probabilistic framework for database selection

In this section, we describe a model which defines a decision-theoretic criterion for optimum database selection ([Fuhr 99]). This model considers relevance as well as other important factors present in distributed retrieval (e.g. costs for query processing and document delivery). We start from the Probability Ranking Principle (PRP, see [Robertson 77]), where it can be shown that optimum retrieval performance is achieved when documents are ranked according to decreasing probability of relevance. Here performance can be measured either in terms of precision and recall (which, in turn, refer to relevance), or by means of a decision-theoretic model which attributes different costs to the retrieval of relevant and nonrelevant documents.

Below, we first describe the basic model for optimum database selection for a given number of documents to be retrieved, thus deriving an optimum selection rule. Then we discuss the consequences of this model for different application situations. In the subsequent section, some related approaches to database selection are described, and it is shown how they fit into the general framework.

5.1 Optimum database selection

In the following, we assume a basic setting as follows: A user submits her query to a broker which has access to a set of IR databases to which it may send the query. In response, each database produces a ranked list of documents, and the broker may request any number of documents from this list; then the user is presented the merged output list. There are database-specific costs for the retrieval of documents, and each database has its own performance curve (e.g. in terms of recall and precision); in addition, the user attributes different costs to relevant and nonrelevant documents presented to her. Now the broker's task is to minimize the expected overall costs by determining the number of documents to be retrieved from each database.

In order to solve this problem, we develop a decision-theoretic model. For this purpose, we assume that we have a probabilistic event space $\mathcal{Q} \times \mathcal{D}$, where $\mathcal{D}$ denotes the set of documents contained in the system, $\mathcal{Q}$ is the set of queries submitted to it; in addition, each pair $(q, d) \in \mathcal{Q} \times \mathcal{D}$ is assigned a relevance judgment. Here we consider each query as a single event, i.e. two users entering the same query formulation are treated as different queries — think e.g. of the large number of one-term queries submitted to Web search engines. Since the system has limited knowledge about queries and documents, it cannot distinguish between queries (or documents) with the same representation, e.g. the same set of terms. Thus, the system is not able to minimize the costs for a single query. Rather, given a query representation, its aim should be to minimize the expected costs for an arbitrary query belonging to this representation.

For the formulation of the decision-theoretic model, we start with the two basic assumptions underlying the PRP, which we extend by an additional one for considering the costs of retrieval in different databases:

1. Relevance judgments are based on a binary relevance scale.
2. The relevance judgment for a document is independent of that for any other document.
3. The costs for retrieving a set of documents from a database are independent of those for other queries or other databases.

The first assumption also can be generalized to multivalued scales, (see [Bookstein 83]). The second assumption not only excludes effects due to similarity or other kinds of dependence between documents, we also ignore the effect of duplicates (i.e. retrieval of the same document from different databases). The third assumption (which we have added to those from the PRP) restricts the nature of the cost factors such that we can regard costs for specific databases and queries in isolation.

In order to estimate costs for a query, we need additional information about the user's standpoint, namely the stopping criterion when looking at the the ranked output list: Does the user want a specific number of documents, or is she looking for a certain number of relevant documents? (There could be even other criteria, e.g. stopping when she has seen a certain number of nonrelevant documents in a row.) As shown in [Fuhr 99], optimum database selection depends on this criterion — unlike the single database case where the PRP tells us that ranking according to decreasing probability of relevance will minimize costs for a variety of stopping criteria. Here we will focus on the number of documents retrieved as stopping criterion.

For retrieving (selecting) s documents from a database D_i, we assume that there is a cost function $C_i^s(s)$, comprising such factors like e.g. connection time, computation costs and charges for delivery. Since the user is interested in finding relevant documents, we attribute a cost factor C^+ to each relevant document retrieved and C^- to each nonrelevant retrieved, with $C^+ < C^-$. For a given query, if we would know the number of relevant documents r that we find among s retrieved from database D_i, then the corresponding costs would be $C_i^s(s) + rC^+ + (s-r)C^-$. However, due to the limited knowledge of the system, it can only estimate the number of relevant documents. For this purpose, let us assume that we know the expected precision $EP_i(s)$ as function of the number of documents selected. Then we arrive at the following formula for the expected costs $EC_i(s)$ for retrieving s documents from database D_i:

$$EC_i(s) = C_i^s(s) + sEP_i(s)C^+ + s(1 - EP_i(s))C^- \qquad (1)$$

Assuming that we have l different databases, and a corresponding vector $\mathbf{s} = (s_1, \dots, s_l)$ of numbers of documents to be retrieved from each of them (with $s_i \geq 0$ for $i = 1, \dots, l$), we can estimate the overall expected costs as the sum of the expected costs for the single databases:

$$EC(\mathbf{s}) = \sum_{i=1}^{l} EC_i(s_i) \qquad (2)$$

Now we can formulate the optimum selection rule: Let $|\mathbf{s}| = \sum_{i=1}^{l} s_i$. For a given number n of documents to be retrieved, determine $\mathbf{s}$ with $|\mathbf{s}| = n$ such that the expected overall cost $EM(n)$ is minimum, i.e.

$$EM(n) = \min_{\substack{\mathbf{s} \\ |\mathbf{s}|=n}} \sum_{i=1}^{l} EC_i(s_i). \qquad (3)$$

This selection rule is rather general, it is only based on the three assumptions from above and holds when the user specifies the number of documents to be retrieved.

5.2 Discussion

Here we discuss the consequences of the selection rule (3).

In order to motivate the subsequent considerations, let us ignore for a moment that $EC(\mathbf{s})$ is a discrete function and assume it to be continuous (i.e. we would allow fractions of documents to be retrieved). Using Lagrange multipliers for specifying the criterion function, we find out that for the optimum solution the cost differentials $\partial EC_i(s_i)/\partial s_i$ are equal for all i with $s_i > 0$ (see e.g. figure 1).

Now we return to the discrete case where we can retrieve whole documents from a database only. Let

$$\Delta_{i,k} = \begin{cases} EC_i(k) - EC_i(k-1) & \text{if } k > 0 \\ 0 & \text{if } k = 0 \end{cases}$$

Table 1: Notations used in this section

symbol	meaning
D_i	database
r	# relevant documents retrieved
s	# documents retrieved (selected)
$\mathbf{s}$	vector $(s_1, \ldots, s_l)$ of # docs to be retrieved from $D_1, \ldots, D_l$
$C_i^s(n)$	costs for selecting n documents from D_i
C_i^0	fixed costs for query processing in D_i
C_i^d	costs for retrieving a document from D_i
C^+	user costs for viewing a relevant document
C^-	user costs for viewing a nonrelevant document
$EP_i(s)$	expected precision when selecting s documents from D_i
$EC_i(s)$	expected costs for retrieving s documents from D_i
$EC(\mathbf{s})$	expected costs for retrieval of $\mathbf{s}$ documents from $D_1, \ldots, D_l$
$EM(n)$	min. expected overall costs for n documents
$P_i(R)$	recall-precision function for database D_i
R_i	# relevant documents in D_i
q	query
d	document
t	term
u_{jm}	indexing weight of term t_j in document d_m
v_j	sum of indexing weights for term t_j in database
w_j	search term weight of term t_j
$\|D_i\|$	size of database D_i (# documents)

denote the cost for retrieving the kth document from database D_i. Obviously these incremental cost differences cannot always be equal for all databases contributing to an optimum solution. The example in table 2 shows that e.g. for $n = 4$, we have $\Delta_{1,2} = 4$ and $\Delta_{2,2} = 2$ for the optimum solution $\mathbf{s}_{\text{opt}}$. As a discrete approximation to equal cost differentials, we define the concept of a uniform vector:

Let $\Delta_{\max}(\mathbf{s}) = \max_i \Delta_{i,s_i}$. Then we call $\mathbf{s}$ a *uniform vector* for a set of databases if the following holds:

$$\forall i : \Delta_{i,s_i} = \Delta_{\max}(\mathbf{s}) \vee \Delta_{i,s_i+1} \geq \Delta_{\max}(\mathbf{s}) \qquad (4)$$

Then we can show the following correspondence between uniform vectors and optimum solutions: For a given set of databases $DB = \{D_1, \ldots, D_l\}$, a given number n of requested documents and any vector $\mathbf{s}$ (with $|\mathbf{s}| = n$) yielding minimum expected overall costs, there exists a uniform vector with the same costs.

Unfortunately, the reverse is not true in the general case, i.e. not every uniform vector yields minimum costs. Thus, let us consider a specific but realistic sub-case, namely that the expected costs for each database are monotonously increasing. We call a set of databases *cost-monotonic*, if for all queries: $\forall i \; \forall k{>}0 \; \Delta_{i,k} \leq \Delta_{i,k+1}$. However, for many applications, the costs for the first document will be higher than the additional costs for the next document, due to the query processing costs. Thus, we call a system *weakly cost-monotonic*, if for all queries: $\forall i \; \forall k{>}1 \; \Delta_{i,k} \leq \Delta_{i,k+1}$. The latter property usually holds when the incremental processing costs per document are constant, i.e. (for $s > 0$)

$$C_i^s(s) = C_i^0 + sC_i^d, \qquad (5)$$

where C_i^0 denotes the query processing costs for database D_i and C_i^d are the additional costs per document delivered. Since users prefer relevant documents ($C^+ < C^-$), according to eqn (1) the assumption is fulfilled when the retrieval quality is monotonically dropping, i.e.

$$\forall i \; \forall s{>}0 \; EP_i(s) \geq EP_i(s+1). \qquad (6)$$

Thus, only when we can ignore the query processing costs, cost-monotonicity seems to be a reasonable assumption. For this case, it can be shown that any uniform vector yields an optimum solution.

We have seen how uniform vectors as discrete approximation to equal cost differentials relate to optimum solutions. Unfortunately, the last statement is only applicable when we can assume cost-monotonicity, which is not realistic in most settings.

Let us return to the assumption of a continuous function $EC(\mathbf{s})$ again, bearing in mind that this is an approximation to the real situation. Furthermore, let us assume that we have weakly cost-monotonic databases where assumptions (5) and (6) hold. Thus, eqn (1) can be rewritten as

$$EC_i(s) = C_i^0 + s(C_i^d + C^-) - sEP_i(s) \cdot (C^- - C^+) \qquad (7)$$

Then we get for the cost differential:

$$\frac{\partial EC_i}{\partial s} = C_i^d + C^- - EP_i(s) \cdot (C^- - C^+) \qquad (8)$$

Since the retrieval quality is assumed to be monotonically decreasing, we see that for large s, when the expected precision is almost zero, the slope of the expected cost function is approximating $C_i^d + C^-$, i.e. the costs per nonrelevant document.

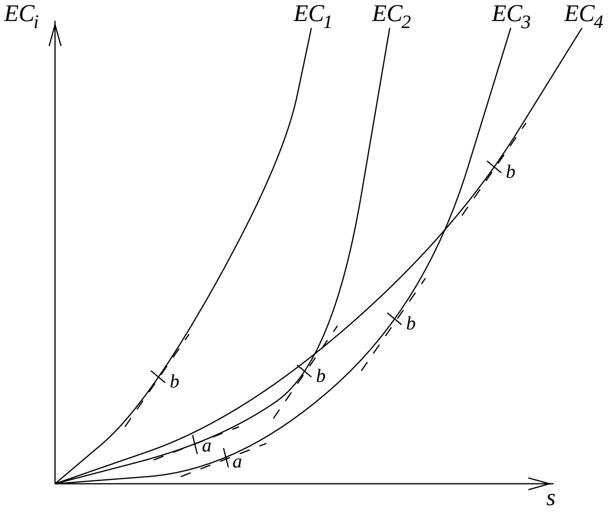

Fig. 1: Sample expected cost functions for $C_i^0 = 0$ with optimum solutions a, b

Now we consider some sub-cases of the general case of a linear cost structure and monotonically decreasing retrieval quality, depending on the structure of the cost factors:

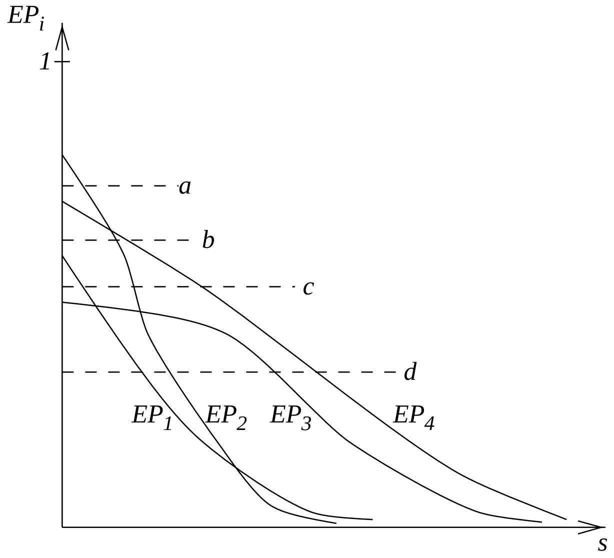

Fig. 2: Sample expected precision curves with optimum solutions $a, \ldots, d$

1. $C_i^0 = 0$ for $i = 1, \ldots, l$: In this case, we have cost-monotonic databases. Sample functions are depicted in figure 1. Since a specific number of total documents retrieved implies an equal slope $\partial EC_i(s_i)/\partial s_i$ for all curves, all databases for which there is a point with this slope on the curve will contribute to the optimum solution. In figure 1, the points corresponding to two solutions a and b are marked, showing that for the first solution, only two of the databases are involved. The set of databases involved grows as the total number of relevant documents increases; a database contributing to a small number always will stay involved for larger numbers, too. This feature is important for incremental retrieval where a user specifies neither the number of documents cost nor the maximum overall costs in advance.

2. $C_i^0 = 0$ for $i = 1, \ldots, l$ and $C_1^d = \ldots = C_l^d$: When also the costs per document retrieved are equal for all databases, then there is a direct relationship to retrieval quality. From eqn (8) it follows that for the cost differentials being equal, also the expected precisions must be equal in this case. In other words, the databases contributing to the optimum solution operate at the same precision level. Figure 2 shows the points for four different solutions $(a, \ldots, d)$, where e.g. for b, only databases 2 and 4 reach this precision level.

3. $C_i^0 > 0$ for some $i \in [1, l]$. If there are databases with nonzero query processing costs, then the set of databases that actually contribute to the solution will depend on the total number n of relevant documents. Here databases involved for small values of n may not contribute to the optimum solution as n grows (see the example in table 2). With regard to incremental retrieval, we have a conflict here: Given that the user first requested n_1 documents and then another n_2 documents, the minimum expected costs for this stepwise procedure may be higher than for retrieving $n_1 + n_2$ relevant documents at once.

Table 2: Example of minimizing expected costs for two weakly cost-monotonic databases

k	$EC_1(k)$	$\Delta_{1,k}$	$EC_2(k)$	$\Delta_{2,k}$	$\mathbf{s}_{\text{opt}}(\mathbf{k})$	$EM(k)$
1	6	6	7	7	$(1,0)$	6
2	10	4	9	2	$(0,2)$	9
3	16	6	14	5	$(0,3)$	14
4	22	6	20	6	$(2,2)$	19
5	28	6	26	6	$(2,3)$	24

5.3 Towards application

With the derivation of the overall cost function $EM(n)$ in eqn (3), we have defined a rule for optimum database selection. So each method for database selection should aim at approximating this optimum. In most applications, it will be difficult to estimate the parameters occurring in $EC_i(n)$. This situation is similar to (or even worse than) the difficulties with the PRP, where the estimation of the probability of relevance of a document also poses problems. However, with our model in mind, we can start with crude approximations of the parameters, apply it and then check how far our estimates deviate from the real parameters — thus telling us where we should spend effort for improving our methods.

In the following, we describe a general procedure for applying our model. In principle, the following steps have to be performed:

1. For each database D_i, estimate the expected precision function $EP_i(s)$ for $s = 1, \ldots, n$.

2. Compute the database-specific cost functions $EC_i(n)$ for $s = 1, \ldots, n$.

3. Derive the optimum cost function $EM(n)$. (Algorithms for this step are given in[Fuhr 99].)

For the estimation of the expected precision, we propose to start from the expected recall-precision-curve. Since evaluations of IR systems typically use this curve, we deem it reasonable to use it also as a starting point. The underlying assumption is that, expected recall-precision varies less than expected precision as function of the number of documents retrieved, and so evaluation studies average over recall, not over number of documents retrieved. On the other hand, little is known about the behavior of query-specific RP curves, and empirical data like e.g. the results from the TREC conference [Harman 95] indicate that there is a great variation in these curves, so further research will be required for achieving good estimates for recall-precision curves.

For the time being, heuristic methods will have to be applied instead. For example, a simple assumption would be a linear function, with $P(0) = P^0$ and $P(1) \approx 0$, thus leading to the approximation $P \approx P^0(1 - R)$; here only P^0 has to be chosen. In the absence of any query-specific knowledge, one might assume that the RP function is equal for all queries. However, in some cases additional information may be available. In practical applications with a set of heterogeneous databases, very often a query contains a condition which cannot be evaluated by the IR system running a specific database; then already $P(0)$ will be very low (see [Fuhr 96]). It also may be feasible to assume functions that are typical for certain kinds of IR systems, e.g. Boolean vs. probabilistic systems.

Given the recall-precision function $P_i(R)$ for a database D_i, we need the expected number of relevant documents R_i in D_i for deriving $EP_i(s)$. Let r denote the number of relevant documents retrieved, then we have $EP_i(s) = r/s = P_i(r/R_i)$.

For the estimation of the number of relevant documents in a database, a simple method can be based on a linear retrieval function that is derived from the uncertain inference view of information retrieval [Rijsbergen 86]. Thus, for a query q, the probability $\Pr(\text{rel}|q,d)$ of a document d being relevant can be formulated based on the probability of the (uncertain) implication $\Pr(q \leftarrow d)$:

$$\begin{aligned}\Pr(\text{rel}|q,d) &= \Pr(\text{rel}|q \leftarrow d) \cdot \Pr(q \leftarrow d) + \\ &\quad \Pr(\text{rel}|\neg(q \leftarrow d)) \cdot \Pr(\neg(q \leftarrow d)) \\ &\approx \Pr(\text{rel}|q \leftarrow d) \cdot \Pr(q \leftarrow d)\end{aligned}$$

The probability that a random document from a database D implies q is

$$\begin{aligned}\Pr(q \leftarrow d | d \in D) &= \sum_{d \in D} \Pr(d) \cdot \Pr(q \leftarrow d) \\ &= \frac{1}{|D|} \sum_{d \in D} \Pr(q \leftarrow d).\end{aligned}$$

We estimate the expected number of relevant documents in D:

$$E(\text{rel}|q,D) \approx \Pr(\text{rel}|q \leftarrow d) \sum_{d \in D} \Pr(q \leftarrow d)$$

The most widely used type of retrieval function is the linear one (see e.g. [Turtle & Croft 91], [Wong & Yao 95]). For this case, the last equation can be simplified further.

$$\begin{aligned}\Pr(q \leftarrow d_m) &= \sum_{t_j \in q} \Pr(q \leftarrow t_j) \Pr(t_j \leftarrow d_m) \qquad (9) \\ &= \sum_{t_j \in q} w_j u_{jm} \qquad (10)\end{aligned}$$

Here $w_j = \Pr(q \leftarrow t_j)$ denotes the search term weight of term t_j and $u_{jm} = \Pr(t_j \leftarrow d_m)$ is the indexing weight of t_j in document d_m.

Thus, the probability that a random document in D implies q can be computed as follows:

$$\begin{aligned}\Pr(q \leftarrow d | d \in D) &= \sum_{d_m \in D} \frac{1}{|D|} \sum_{t_j \in q} w_j u_{jm} \qquad (11) \\ &= \frac{1}{|D|} \sum_{t_j \in q} w_j \sum_{d_m \in D} u_{jm} \\ &= \frac{1}{|D|} \sum_{t_j \in q} w_j v_j\end{aligned}$$

(where $v_j = \sum_{d_m \in D} u_{jm}$).

The expected number of relevant documents in D can be approximated by

$$E(\text{rel}|q,D) \approx \Pr(\text{rel}|q \leftarrow d) \cdot \sum_{t_j \in q} w_j v_j \qquad (12)$$

Unless we have query-specific relevance feedback data, we can only assume a global constant c for estimating $\Pr(\text{rel}|q \leftarrow d)$. Now we can approximate the number R_i of relevant documents in the database D_i for the actual query as follows:

$$R_i \approx c \sum_{t_j \in q} w_j v_j. \qquad (13)$$

This formula for estimating the number of relevant documents has the same structure as the retrieval function (10). However, whereas a retrieval function computes the retrieval status value for each document, formula (13) yields the expected number of relevant documents for each database, i.e. the estimation procedure treats databases like meta-documents. For applying this formula, a broker would need only the parameter v_j for each term occurring in a database. Thus, by applying the same data structures and algorithms as for ordinary retrieval, formula (13) can be evaluated rather efficiently, even for a large number of databases.

6 Database selection — models and experimental results

Although the theoretical approach presented above forms a general framework for database selection, it does yield a model for actually solving this task. The relationship between this framework and database selection models is the same as that between the PRP and probabilistic retrieval models. Fortunately, a number of database selection models has been developed already independently of the framework presented here. Now, we will show how these models fit into the framework.

The Gloss system described in [Gravano et al. 94] and [Gravano & Garcia-Molina 95] takes a heuristic approach towards database selection. Based on the vector space model, two additional assumptions are made:

1. All databases use the same retrieval function for computing retrieval status values (RSV).
2. Given a query q and a document d, the document is only useful (potentially relevant) for q if its RSV exceeds a certain threshold l.

Then a measure of goodness for a database D with respect to a query q and a cutoff value l is defined:

$$\mathit{Goodness}(l,q,D) = \sum_{d \in D \wedge \mathit{sim}(q,d) > l} \mathit{sim}(q,d) \qquad (14)$$

In addition, different assumptions about the distribution of term weights within a database can be made, namely either high positive or high negative correlation of different terms. However, for a cutoff value $l = 0$, the *Goodness* measure is the same in both cases. If document term weights and search term weights follow the probabilistic interpretation as shown in eqn (10), the *Goodness* measure corresponds to the estimated number of relevant documents in a database according to eqn (13) (with $c = 1$). In [French et al. 98], a relevance based evaluation of the Gloss method is performed. It turns out that Gloss performs well at predicting the distribution of RSVs within databases; however, with respect to retrieval quality (i.e. selecting those databases that contain many relevant documents), the performance is only moderate.

A similar result is described in [Gövert 97], where the estimation formula (12) is investigated. It turns out that in principle (for the small set of databases used in this study), this formula gives good approximations of the number of relevant documents; however, the factor $\Pr(\text{rel}|q \leftarrow d)$ is highly query-dependent, and there is no obvious method for estimating this parameter.

The CORI model presented in [Callan et al. 95] performs database selection based on a new collection ranking formula for databases; this formula is similar to document retrieval based on *tf* · *idf* weighting, but treats collections like documents. The belief $P(t_i|D)$ in collection D with respect to term t_i is determined by:

$$\begin{aligned} T &= \frac{f_i}{f_i + 50 + 150 \cdot s(D)/\bar{s}} \\ I &= \frac{\log\left(\frac{l+0.5}{g_i}\right)}{\log(l+1)} \\ P(t_i|D) &= 0.4 + 0.6 \cdot T \cdot I \end{aligned}$$

where:

f_i number of documents in collection D containing t_i,

g_i number of collections containing t_i,

l number of collections to be ranked,

$s(D)$ number of words in D,

$\bar{s}$ mean of $s(D)$ for the collections to be ranked.

The overall weight of a collection D w.r.t. a query q depends on the query structure, but is usually just the average of the $P(t_i|D)$ values for the query terms. Thus, this function is similar to our weighting formula (13), but uses a different weighting scheme for the terms in a collection. In [Xu & Callan 98], this approach is extended to phrase information and query expansion, and it is shown that these techniques improve the outcome of the database selection process.

In [French et al. 99], a comparative study of the database selection methods of Gloss and CORI is presented, where CORI clearly outperforms Gloss. A detailed analysis of results shows that the major weakness of Gloss is the fact that it does not distinguish between a database with many marginal relevant documents and another one with a few highly relevant documents in case the sums of RSVs are the same. With respect to our general framework, we also would assume that the expected number of relevant documents is the same, but we could handle this problem by assuming different recall-precision functions, where the database with the marginal relevant documents yields a lower retrieval quality and thus the other database should be preferred.

A probabilistic model for database selection and data fusion is presented in [Baumgarten 97] and [Baumgarten 99]. Based on two standard probabilistic retrieval models for the non-distributed case (namely the binary independence retrieval model ([Robertson & Sparck Jones 76])and the retrieval model with probabilistic indexing ([Fuhr 89]), distributed versions of these models are developed. Then database selection is performed based on the expected distributions of RSVs. Due to the log-linear structure of these retrieval models, there is no direct correspondence to the number of relevant documents or cost parameters of our general framework. However, this approach has the advantage that it is based on a well-founded model for distributed retrieval, whereas other approaches are more heuristic.

All these approaches (as well as the estimation formula (13) derived in the previous section) perform database selection by using a term-wise weighting formula for databases. As an alternative approach, two query-based strategies are described in [Voorhees et al. 95]. Both of these strategies consider similarities between queries. In the first case, similarity is based on term-wise comparison of queries, and then relevance feedback information from the most similar past queries is used for database selection. The second method first performs query clustering based on the sets of retrieved documents from the different databases. By averaging the query vectors of a cluster, the centroid vector is formed; for a new query, first the most similar centroid vectors are determined, and then relevance feedback information is used for database selection as before.

A system-oriented aspect of database selection is investigated in [Dushay et al. 99], namely response time of database servers. It is shown how response time statistics from the past can be used for predicting response time for the current query. In terms of our general framework, response time could be modeled as cost factor that contributes to query processing costs, and thus both retrieval quality and response time could be considered for database selection.

7 Collection fusion

Once the databases to be used for answering a query have been selected, the query can be sent to these databases in order to compute the answers. When the answers are received by the site initiating the query (or an intermediate broker), they have to be merged in order to produce a single result. The final retrieval quality depends heavily on this collection fusion step. Theoretically, the PRP gives a hint how this task should be performed — namely by ranking the documents from the union of all answers according to decreasing values of their probability of relevance. However, since the single databases have no or little knowledge about the overall term statistics (i.e. global *idf* values), the RSVs of these databases are poor predictors of the probability of relevance.

The distributed retrieval model in [Baumgarten 99] avoids this problem by collecting term statistics from all databases and propagating the global *idf* values to the selected databases. The STARTS protocol proposed in [Gravano et al. 97] also is based on global *idf* values; instead of propagating these to the databases, however, the databases are requested to transmit the document term weights for each element of the answer set, so that a recomputation of RSVs based on the global *idf* values can be performed.

In [Callan et al. 95], four different strategies for collection fusion are investigated:

interleaving takes documents from the different ranked answer list in a round-robin fashion, i.e. only the rankings, but not the RSVs of the single databases are taken into account.

raw score merge uses the RSVs from the different databases and merges according to descending RSVs.

normalized merge modifies the original RSVs by considering the collection weights from the database selection process, before merging according to these "normalized" RSVs.

weighted merge uses global *idf* weights for computing the RSVs on which the merging process is based.

As expected, *interleaving* performs worst, and *weighted merge* gives the highest retrieval quality. *Raw score merge* also gives poor results. Fortunately, the outcome of *normalized merge* is close to that of *weighted merge*; since in this case, no global *idf* values have to be considered (and thus the corresponding communication and computation overhead is avoided), this method seems to be most appropriate for performing collection fusion.

8 Conclusions and outlook

In this paper, we have given a survey on research problems related to resource discovery. We have shown that the diversity of users, information needs, media, indexing schemes, document formats, database schemas, types of services, systems, protocols and other parameters pose a large number of problems for which only partial solutions will be available in the near future.

For the specific problems of database selection and collection fusion, we have described the approaches that are currently available. Relating these solutions to the dimensions of the problem space, it turns out that only a few of these parameters have been considered yet, thus these solutions are applicable in very limited domains only. On the other hand, these approaches may serve as starting points for the development of more complete models.

Interaction with existing global search services for the Web leads to the misconception that the future of distributed search simply involves improvements in quality of response to the list of keywords in a query. While improvements in precision and recall, for example, are important, efficient and effective distributed search will potentially enable the construction of entirely new classes of information-based applications. New kinds of information, new forms of user interaction, and new business models place entirely different demands on distributed search technology.

Over the next decade, today's information applications will experience significant evolution as technology improvements lead to ubiquitous access to networks, improvements in bandwidth and reliability, and increase in the amount of quality information available.

Today's information-based applications rely on static document collections. Tomorrow's applications will build and maintain dynamic collections of networked documents that adapt to changing requirements, technology, and availability of information. Embedded within these applications will be complex, domain-specific, distributed information retrieval software that will find, filter, sort, and present information relevant to the topic of interest. When the information is the application, the underlying technology must adapt.

References

Baumgarten, C. (1997). A Probabilistic Model for Distributed Information Retrieval. In: Belkin, N.; Narasimhalu, D.; Willet, P. (eds.): *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 258–266. ACM, New York.

Baumgarten, C. (1999). A Probabilistic Solution to the Selection and Fusion Problem in Distributed Information Retrieval. In [SIGIR 99].

Bookstein, A. (1983). Outline of a General Probabilistic Retrieval Model. *Journal of Documentation 39(2)*, pages 63–72.

Callan, J.; Lu, Z.; Croft, W. (1995). Searching Distributed Collections with Inference Networks. In [Fox et al. 95], pages 21–29.

Connolly, D. (ed.) (1997). *XML: Principles, Tools, and Techniques*, volume 2 of *World Wide Web Journal*. O'Reilly, Sebastopol, California.

Croft, W. B.; Moffat, A.; van Rijsbergen, C. J.; Wilkinson, R.; Zobel, J. (eds.) (1998). *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York. ACM.

Dushay, N.; French, J.; Lagoze, C. (1999). Predicting Indexer Performance in a Distributed Digital Library. In: *Research and Advanced Technology for Digital Libraries*. Springer, Berlin et al.

Fox, E.; Ingwersen, P.; Fidel, R. (eds.) (1995). *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York. ACM.

French, J.; Powell, A.; Viles, C.; Emmitt, T.; Prey, K. (1998). Evaluating Database Selection Techniques: A Testbed and Experiment. In [Croft et al. 98], pages 121–129.

French, J.; Powell, A.; Callan, J.; Viles, C.; Emmitt, T.; Prey, K.; Mou, Y. (1999). Comparing the Performance of Database Selection Algorithms. In [SIGIR 99].

Fuhr, N. (1989). Models for Retrieval with Probabilistic Indexing. *Information Processing and Management 25(1)*, pages 55–72.

Fuhr, N. (1996). Object-Oriented and Database Concepts for the Design of Networked Information Retrieval Systems. In: Barker, K.; Özsu, M. (eds.): *Proceedings of the Fifth International Conference on Information and Knowledge Management*, pages 164–172. ACM, New York.

Fuhr, N. (1999). A Decision-Theoretic Approach to Database Selection in Networked IR. *ACM Transactions on Information Systems 17*. (To appear).

Gövert, N. (1997). Evaluierung eines entscheidungstheoretischen Modells zur Datenbankselektion. In: Fuhr, N.; Dittrich, G.; Tochtermann, K. (eds.): *Hypertext — Information Retrieval — Multimedia (HIM). Theorien, Modelle und Implementierungen integrierter elektronischer Informationssysteme*. Universitätsverlag Konstanz. http://ls6-www.cs.uni-dortmund.de/~goevert/HIM97/.

Gravano, L.; Garcia-Molina, H. (1995). Generalizing GlOSS to Vector-Space Databases and Broker Hierarchies. In: Dayal, U.; Gray, P.; Nishio, S. (eds.): *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases*, pages 78–89. Morgan Kaufman, Los Altos, California.

Gravano, L.; Garcia-Molina, H.; Tomasic, A. (1994). The Effectiveness of GlOSS for the Text Database Discovery Problem. In: Snodgrass, R. T.; M., W. (eds.): *Proceedings of the 1994 ACM SIGMOD. International Conference on Management of Data.*, pages 126–137. ACM, New York.

Gravano, L.; Chang, C.-C.; Garcia-Molina, H.; Paepcke, A. (1997). STARTS: Stanford Proposal for Internet Meta-Searching. In: *Proceedings of the 1997 ACM SIGMOD International Conference On Management of Data*. ACM, New York.

Harman, D. (1995). Overview of the Second Text Retrieval Conference (TREC-2). *Information Processing and Management 31(03)*, pages 271–290.

Miller, E. (1998). An Introduction to the Resource Description Framework. *D-Lib Magazine 4(5)*. http://www.dlib.org/dlib/may98/miller/05miller.html.

van Rijsbergen, C. J. (1986). A Non-Classical Logic for Information Retrieval. *The Computer Journal 29(6)*, pages 481–485.

Robertson, S.; Sparck Jones, K. (1976). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science 27*, pages 129–146.

Robertson, S. (1977). The Probability Ranking Principle in IR. *Journal of Documentation 33*, pages 294–304.

SIGIR (ed.) (1999). *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, New York. ACM.

Turtle, H.; Croft, W. (1991). Evaluation of an Inference Network-Based Retrieval Model. *ACM Transactions on Information Systems 9(3)*, pages 187–222.

Voorhees, E.; Gupta, N.; Johnson-Laird, B. (1995). Learning Collection Fusion Strategies. In [Fox et al. 95], pages 172–179.

Weibel, S. (1995). Metadata: The Foundations of Resource Description. *D-Lib Magazine 1(July)*. http://www.dlib.org/dlib/July95/07weibel.html.

Wong, S.; Yao, Y. (1995). On Modeling Information Retrieval with Probabilistic Inference. *ACM Transactions on Information Systems 13(1)*, pages 38–68.

Xu, J.; Callan, J. (1998). Effective Retrieval with Distributed Collections. In [Croft et al. 98], pages 112–120.