

# Ensuring Retrieval Effectiveness in Distributed Digital Libraries

JAMES C. FRENCH<sup>1</sup> AND CHARLES L. VILES<sup>2</sup>

*Department of Computer Science, University of Virginia, Charlottesville, Virginia 22903*

Received August 21, 1995; accepted December 4, 1995

---

We find that dissemination of collection-wide information (CWI) in a distributed collection of documents is needed to achieve retrieval effectiveness comparable to that of a centralized collection. Complete dissemination is unnecessary. The required dissemination level depends upon the content-skew of the distributed collection, i.e., how documents are allocated among sites. Low dissemination is needed for random document allocation, but higher levels are needed when documents are allocated on the basis of their content. We define parameters to control dissemination and document allocation and present results from four document collections. These results provide insight into the necessary technology underlying digital libraries. We also describe the architecture of the Networked Computer Science Technical Report Library (NCSTR), a concrete example of a system that fits our model of a distributed archive. © 1996 Academic Press, Inc.

---

## 1. INTRODUCTION

Interest in the concept of digital libraries has exploded in recent years, fueled by the speculation and promise of the “information superhighway.” Even though there is an intuitive appeal to the notion, it remains for the present vaguely defined. For the purpose of this paper we regard a digital library as a coherent collection of digital materials which is organized to provide users with efficient and effective access to those materials. A number of workshops have recently focused attention on digital library issues. Some have considered the issues very broadly [1] while others have attempted only to define specific research areas [2].

Although there are many economic, social, and legal issues to be resolved before digital libraries can become widespread, this paper focuses on some of the technological issues. There are many technological problems seeking solutions, such as:

- object encapsulation and long-term (indefinite) archiving;

<sup>1</sup> E-mail: french@virginia.edu.

<sup>2</sup> E-mail: viles@virginia.edu.

- collection management;
- organizing and indexing the materials for storage and retrieval;
- user interfaces and human-computer interaction; and
- interoperability of disparate heterogeneous systems.

We concentrate here on organizational aspects of a digital library that bear on the retrieval performance of the system. From that perspective it is useful to consider the digital library as a distributed information retrieval (IR) system. As in traditional centralized IR systems, the performance of such systems is gauged along two dimensions: *efficiency*—how quickly a query is processed; and *effectiveness*—the quality of the resulting response. In this paper, we are concerned with the latter dimension, retrieval effectiveness. How can we achieve effectiveness comparable to that of a static, centralized archive in an environment that is inherently distributed and dynamic? Most advanced IR models rely on information gathered from the entire collection of documents to aid in the retrieval process. In a distributed, dynamic environment, this *collection-wide information* (CWI) is constantly changing as new documents are added. However, it is not clear how often member sites of a distributed archive should disseminate the knowledge of new document insertions to other sites, or even if such dissemination is necessary. In this work we consider the level at which CWI needs to be maintained at each member site in order to ensure retrieval effectiveness commensurate with a centralized archive. Our contributions include:

- a model for CWI dissemination within a distributed collection of documents;
- a finding that little if any dissemination is needed for distributed collections where documents are randomly allocated to sites; and
- a finding that higher dissemination levels are needed when documents are allocated to sites so that similar documents are collocated.

To set the appropriate context, we begin this article with a description of a concrete example of a deployed digital library, NCSTR, the Networked Computer Science Technical Report Library. We follow with related work. We

then describe the distributed archive and provide a description of CWI dissemination, document allocation, and the parameters we use to model these attributes. We continue with a description of our experiments. Our results and a detailed discussion of some of the issues and questions raised by our work follow. We finish with a summary and some directions for future work.

## 2. NCSTRL: A DIGITAL LIBRARY OF TECHNICAL REPORTS

NCSTRL is the outgrowth of an ongoing collaboration among researchers at Cornell University, Old Dominion University, Stanford University, SUNY Buffalo, University of Virginia, and Virginia Polytechnic Institute and State University. In November 1995 NCSTRL was moved from prototype status to a fully operational system. At that time it was the largest (by “largest” we mean that NCSTRL contained 30 participating distributed sites when it became operational and that number is growing) operational digital library system in use. This section briefly describes the architecture of that system at a level that is appropriate to motivate the investigation that is the main topic of this paper.

Over the past several years a number of efforts have been directed toward providing access to computer science technical reports over the Internet. Chief among these technical report servers accessible via the World-Wide Web were the Unified Computer Science Technical Report Index (UCSTRI) [3] at the University of Indiana, Harvest [4], the CS-TR project which developed Dienst [5, 6], the WATERS (Wide Area TEchnical Report Service) project [7, 8], and the precursor to WATERS, *techrep* [9]. Dienst and WATERS are described more fully in a special issue of the *Communications of the ACM* [10]; together they form the basis of the NCSTRL architecture.

All these experiments have been successful and have paved the way for NCSTRL, the Networked Computer Science Technical Report Library. UCSTRI showed that a sufficient volume of resources was available online (via anonymous FTP) to be useful to researchers. WATERS improved the level of service by adopting a uniform file format for cataloging and indexing papers, and by providing tools to simplify library maintenance. As a result, the user sees a more uniform collection than with UCSTRI. Dienst adopted an open architecture for its distributed library, supporting multiple document formats and a more flexible searching system.

NCSTRL provides a single uniform computer science technical report library with a collection built from the technical reports of an international group of computer science and engineering universities and research laboratories. Such a collection benefits the researchers who consult it and the departments that contribute to it. Research-

ers are able to easily search a body of material that is now slow and difficult to access. Departments have a clean, effective management system for their technical reports and eliminate much of their current copying and mailing charges.

The NCSTRL architecture combines the power and flexibility currently found in Dienst with the ease of installation of WATERS. NCSTRL software comes in two levels, Lite and Standard. The Lite level, based on the current WATERS package, has a lower startup investment, while the Standard level, based on Dienst, offers greater functionality for a larger investment. There is a uniform user interface providing access to all the reports in the system.

The architectural details relevant to this paper concern the distribution of the holdings. This is described in more detail below. For more details of the technology underlying NCSTRL the reader should consult Davis and Lagoze [11], Lagoze *et al.* [12], and French [9], all available via NCSTRL (<http://www.ncstrl.org/>).

### 2.1. NCSTRL Architecture

Each NCSTRL-Standard site is logically composed of a User Interface (UI) server, an Index server, and a Repository (Fig. 1). A fourth, centralized server, the Connections server, provides each site with the addresses of all other servers. NCSTRL users interact only with the UI server, which mediates all access to the system. The UI server communicates with other servers using the Dienst protocol [11], which, though it employs HTTP as a transport layer, is intended for use by programs, not users, and thus provides an open architecture. NCSTRL index servers differ from WATERS in that NCSTRL index servers are distributed, not centralized, and return results expressed in the Dienst protocol. These results are transformed by the UI server into a form suitable for display to the user.

The NCSTRL-Lite sites are handled differently. There is a single, possibly replicated, NCSTRL index server at which all the NCSTRL-Lite sites register their bibliographic information. The repositories for each NCSTRL-Lite site are managed locally; that is, the holdings are distributed but the index is centralized. To the larger system this centralized index server looks like a single NCSTRL-Standard site (Fig. 2).

All documents in NCSTRL have a unique, location independent name, the *docid*, and can exist in multiple formats. The repository stores the various formats. To conduct a search, the user fills out a form; the UI server then queries each site's index server in parallel, each of which returns a list of docids for matching documents. Selecting from this list causes the UI to show a list of available formats of the document, as reported by the repository. Selecting one of these formats causes the repository to retrieve the document or document portion.

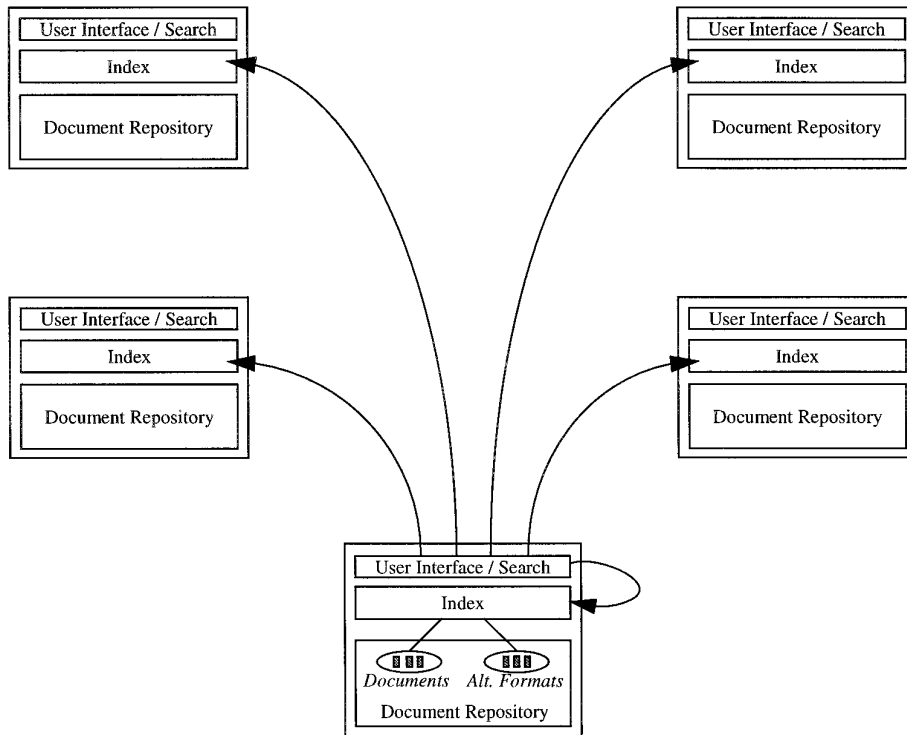


FIG. 1. NCSTRL-Standard architecture. Each site has three servers: User Interface (UI), Index, and Repository. The UI server searches all Index servers in parallel. Each document has a unique name and may exist in many formats.

This architecture provides enough flexibility to configure the system in a variety of ways. The logical function would be the same for each configuration but the performance characteristics might vary widely. We may have all the repositories distributed, most or all of the indexes distributed, and some or all of the components replicated. The chief question being addressed in the remainder of this

paper is how will the dissemination of collection-wide information in such distributed IR systems affect performance?

As was emphasized in the HPCC/IITA Workshop on Digital Libraries [2], scaling is a major issue in digital library systems. They may be expected to grow to hundreds or thousands, of autonomously managed distributed repositories, and these must be maintained and integrated into a coherent whole. The following sections discuss the effects of dissemination of collection-wide information among the sites of a distributed IR system. This work is part of our larger research program which also considers the effects of update activity on these systems. Studies of this kind are necessary to gain enough information to be able to properly engineer working, high-performance digital library systems such as NCSTRL.

### 3. RELATED WORK

In the previous section we described some of the other on-line technical report services. In this section we briefly describe other research efforts in distributed information retrieval systems.

Callan *et al.* [13] adapted the inference net approach used in the INQUERY system to the identification of collections

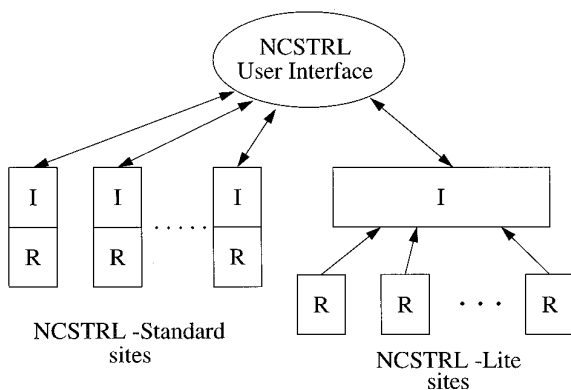


FIG. 2. NCSTRL architecture. Each Standard site has an index (I) and repository (R). Each Lite site is registered with an index server and all the Lite sites together look like a single NCSTRL-Standard site. The index for the Lite sites is centrally located, but the repositories are distributed.

likely to contain relevant documents. A search query was then broadcast to these collections, searches performed, and results merged. Effectiveness was found to be very similar to a “central” collection composed of the documents in the distributed collection. Collection-wide information was assumed to be available. In our work, we do not consider end-to-end issues, but concentrate specifically on the level at which CWI must be maintained to ensure reasonable search quality.

Harvest [4, 14] is a prototype system designed to address some of the problems in resource discovery and information access on the Internet. It includes efficient mechanisms for gathering and indexing topic-specific information at a central location. Mechanisms for caching and replicating the indexes are provided. Harvest concentrates on making efficient use of network resources. Effectiveness considerations are secondary.

In the Parallel InfoGuide system [15], Aalbersberg and Sijstermans use a distributed-memory multi-processor to get very fast query response times. They use the Vector Space Model [16] as the IR engine. To get good effectiveness while retaining ease of updates, inverse document frequency (idf) based term weights are kept with a dictionary and not with the documents. The weights are then applied to the query terms only. This limits the kinds of term weighting functions that may be used by the system. There is no notion of a distributed system with autonomous sites or of lazy dissemination of CWI.

Viles [17] describes a method for maintaining CWI in a distributed IR system. A separate, replicated service maintains the CWI, accepting updates from sites in the system and serving up the latest version of the CWI in reply. However, it is not clear whether this method is sufficient to maintain the retrieval effectiveness of the IR system or if it is overkill. In our work, we concentrate on determining the level of dissemination needed to maintain retrieval effectiveness.

Mazur [18] provides a theoretical treatment of some issues in distributed IR. He showed that a global thesaurus exists for a set of disjoint information systems using boolean retrieval with thesauri. He also showed that each separate site could be considered a simple restriction of a global system.

Harman *et al.* [19] describe a prototype distributed IR system where data is stored centrally but maintained in separate datasets organized by content. Datasets are then cached to machines where extensive access to the data is anticipated. Searches could span multiple datasets kept at multiple locations, but any single dataset was never divided. While CWI was used in the form of idf term weights, since the datasets never spanned multiple locations, no dissemination was needed.

There is also considerable work occurring in methods

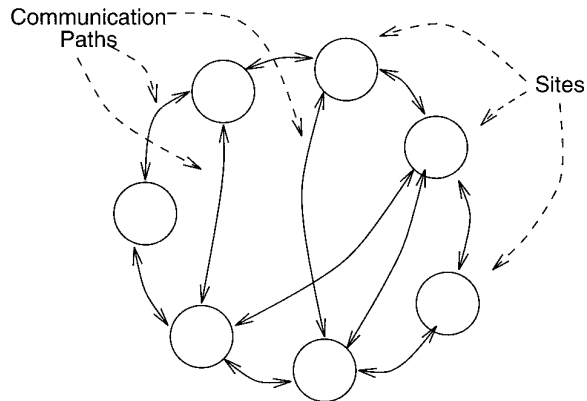


FIG. 3. The topology for an instance of a distributed archive. Document insertions happen asynchronously at each site. Sites communicate with each other to exchange information on their respective collections and to answer queries.

to intelligently merge result lists from separate searches into a single result. The issues and several approaches are well-described in Vorhees *et al.* [20] and Belkin *et al.* [21].

Work by Tomasic and Garcia-Molina [22] is performance-oriented, focusing on distributed index architectures that provide low query response time. Effectiveness is not considered.

#### 4. DISTRIBUTED ARCHIVE MODEL

In a distributed archive (Fig. 3), documents are not kept in a single central location, but are distributed over many sites. A search performed in such an archive must be executed (at least logically) at every site, and the results from each site combined in a meaningful way for presentation to the user. To achieve high effectiveness, sites also communicate with each other to exchange information on their respective collections.

Documents arrive in the system and are allocated to sites based upon some criteria. These criteria may be administrative, e.g., the document was created at site  $i$  so it resides at site  $i$ , or they may be content-based, e.g., the document is similar to some others so it will be collocated with them. This effectively creates a document stream for every site and is the source of insertions into a site’s local collection. In an evaluation environment, the source of the stream is generally a group of documents for which there are accompanying queries and relevance judgments.

In this work we assume sites cooperate with each other. In particular, all sites agree on an information retrieval model. While this assumption finesses interesting problems regarding result list merging, it allows us to concentrate on dissemination intensity issues.

#### 4.1. Dissemination of CWI

Before describing the dissemination model, some notation is needed. The distributed archive is composed of  $s$  sites. The  $j$ th document at site  $i$  is denoted by  $D_{ij}$ . At any site  $i$ , there are two collections represented.  $C_i^l$  represents the ordered collection of documents physically stored at site  $i$ —the “local” collection. The order corresponds to the insertion order of documents at that site.  $C_i^s$  represents the collection of documents that has been used to generate site  $i$ ’s version of CWI. We call this CWI version  $G_i$ , so  $G_i = f(C_i^s)$ .

Most highly effective IR models use information gathered from the entire collection to aid in retrieval. Probably the most wide-spread instance of this collection-wide information is the *inverse document frequency* (idf) defined for all concepts in the collection. It is fair to say that the use of idf information is ubiquitous in research IR systems. In the TREC-1 [23] conference, fully 70% of all contributors used some form of the idf, including the top six performers in the *ad hoc* experiments and the top five in the routing experiments. The idf for the  $k$ th concept or term is given by

$$\text{idf}_k = \log \left( \frac{N}{\text{df}_k} \right), \quad (1)$$

where  $N$  is the total number of documents and  $\text{df}_k$  is the document frequency for the  $k$ th term. Both  $N$  and  $\text{df}_k$  are collection-wide statistics. In the distributed archive, a global idf requires information from all sites, so the above equation now becomes

$$\text{idf}_k = \log \left( \frac{\sum_{i=1}^s N_i}{\sum_{i=1}^s \text{df}_{ik}} \right), \quad (2)$$

where  $N_i$  and  $\text{df}_{ik}$  represent the contribution of each site to the global (or collection-wide) idf, and  $s$  is the number of sites in the archive. Sites must agree on the identity of the  $k$ th term.

The addition of a single document causally affects the CWI. In a completely faithful implementation of an IR model using CWI, this would require dissemination of information from the document insertion to all sites so a consistent idf could be maintained. However, it is not clear that the addition of a single document—or group of documents for that matter—changes the CWI enough to influence the overall effectiveness of the IR system. The goals of an IR system generally do *not* include serializability of updates on the idf, so it may be possible to allow lazy dissemination of document insertions without impairing retrieval effectiveness. At least two questions arise:

- At what intensity does CWI need to be circulated to maintain retrieval effectiveness?

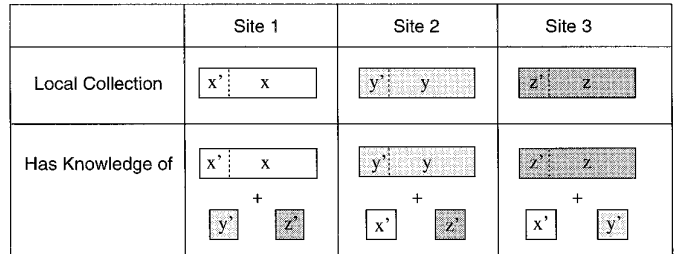


FIG. 4. The degree of dissemination found in a distributed archive with three sites and  $d = 0.25$ . The horizontal blocks represent the stream of documents stored at each site, in the order they were inserted. Each site knows about its own documents and the first 25% of the documents inserted at other sites.

- How should CWI be circulated?

We consider only the first question in this paper as it has a profound influence on the answer to the second question. Check [17] for an algorithm that addresses the second question.

*Dissemination Model.* Let  $\text{prefix}(d, C_i^l)$  be the first  $d$ th fraction of  $C_i^l$ . The parameter  $d$  defines the degree of dissemination of CWI in the archive. At any point in time, site  $i$  knows about all of its own documents plus  $\text{prefix}(d, C_j^l) \forall j \neq i$ . That is,

$$C_i^s = C_i^l \cup \left( \bigcup_{j \neq i} \text{prefix}(d, C_j^l) \right). \quad (3)$$

Note the following about the dissemination parameter:

- $d$  varies continuously between 0 and 1.
- When  $d = 0$ , no dissemination occurs and  $G_i$  is derived solely from local holdings.
- When  $0 < d < 1$ ,  $G_i$  is derived partly from local holdings and partly from documents held elsewhere.
- When  $d = 1$ , complete dissemination occurs. Every site has “perfect” knowledge of every other site. So  $G_i = G_j \forall i, j$ .

Figure 4 illustrates this dissemination for  $d = 0.25$ .

#### 4.2. Allocation of Documents

Documents may be physically allocated among all sites in a variety of ways. At one extreme, the physical location of documents may be completely independent of document content. For example, in a distributed archive of 20th century American Literature, a copy of one of Fitzgerald’s letters might be stored at any location in the archive with equal probability. At the other extreme, a document’s content may be highly correlated with its physical location: in our example, most of Fitzgerald’s correspondence would be stored in the same place, with just a small portion held

```

D = getNextDocFromStream ();
if (relevantQueryForDoc(D) and Bernoulli (a)) {
    Q = findRelevantQuery (D);
    assignedSite = QHome(Q);
} else {
    assignedSite = Equilikely (1, numSites);
}

```

FIG. 5. Pseudo-code for the allocation of documents among all sites. `Bernoulli ( $a$ )` returns true with probability  $a$  and false otherwise. `Equilikely (1, numSites)` returns an integer  $j$  uniformly distributed in  $1 \leq j \leq \text{numSites}$ .

at other sites. One can easily imagine distributed archives where one or the other extreme is realistic.

*Allocation Model.* Qualitatively, we wish to see how varying the allocation of documents to sites affects retrieval performance. Our approach is to assume that documents that are relevant to the same query are relevant to each other. We assign each query  $Q$  a random home site,  $QHome(Q)$ . Documents are assigned to sites based on three types of information:

- relevance information,
- $QHome()$ , and
- an affinity probability  $a$ .

If document  $D$  is relevant to query  $Q$ , then  $D$  is assigned to  $QHome(Q)$  with probability  $a$ , and is assigned at random across all sites with probability  $1 - a$ . This means that  $D$  is assigned to  $QHome(Q)$  with probability slightly greater than  $a$ :  $a + (1 - a)/s$  to be exact. If  $D$  is not relevant to any query, then it is assigned randomly to any site in the archive. This algorithm assumes that documents are not relevant to more than one query: not completely realistic, but reasonable to a first approximation for our purposes. Figure 5 shows pseudo-code for the allocation strategy. Figure 6 shows the probability distribution for the location of document  $D$  given 5 sites,  $D$  is relevant to query  $Q$  and  $QHome(Q)$  is site 2 for  $a = 0.5$  and  $a = 0.0$ .

The attraction of defining the affinity parameter in this manner is that

- when  $a = 0$ , documents are randomly allocated across all sites, mapping to the case where content has nothing to do with document location, and
- when  $a = 1$ , documents relevant to the same query are collocated, mapping to the case where content has a large influence on document location.

We chose to randomly allocate nonrelevant documents because it was the most convenient method consistent with using relevance judgements to determine content-similarity. We also considered allocation based upon the outcome of a document clustering method. This approach has drawbacks in that it introduces an additional variable—the

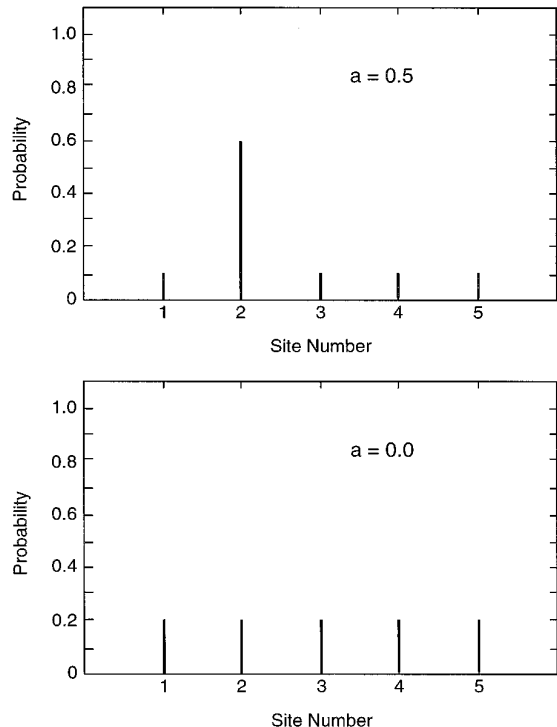


FIG. 6. Probability that a document  $D$  will be assigned to a site given five sites,  $D$  is relevant to  $Q$ ,  $QHome(Q) = 2$ , for affinities of 0.0 and 0.5.

choice of clustering methodology. The method we used is attractive both for its simplicity and because it gets to the heart of measuring effectiveness—the disposition of relevant documents in the distributed system.

Two additional points are important. First, we are defining a model that gives us experimental, parametric control over the degree of *content-skew* in the distributed IR system. We are not recommending a clustering methodology. In fact, in an operational environment, our criterion is problematic because it requires known queries and relevance judgements. Second, while we can control the degree of content-skew in an experimental setting, in a working system with largely autonomous sites, the amount of content-skew is a property of the system and not necessarily under administrative control.

## 5. DESCRIPTION OF EXPERIMENTS

### 5.1. Software Description

The software we use to run our dissemination and allocation experiments is called DRIFT. DRIFT is an object-oriented implementation of the Vector Space Model [16] written in C++ and designed specifically to perform experiments in distributed IR.

There are several fundamental objects in DRIFT, but we

describe only the two most important ones here. A site object maintains its own document collection and view of the CWI. A Site also maintains a list of possible search functions to use (essentially different term weighting strategies and a dissemination policy associated with the local collection).

The second important object is the DocStream. The DocStream object serves as a source of documents to the IR system. Logically, there is a DocStream associated with every site. In our implementation, we have a single DocStream associated with a master site. The master site splits its stream into multiple streams for each site according to whatever allocation scheme is in place. The source for a DocStream can be any collection of documents, though in order to perform evaluation experiments, the collection must have an accompanying set of queries and relevance judgements.

Besides controlling the document stream, the master site initiates all searches and then collates and evaluates the search results. It also can configure each Site object so they all use a common dissemination policy and term weighting strategy.

DRIFT is explicitly instrumented to do search and evaluation at intermediate points in a stream of document insertions. This enables users to add a dynamic component to the distributed archive and to monitor effectiveness in an evolving collection. In the experiments reported here, we did not use the dynamic capability of DRIFT. A single evaluation was performed at the end of the experiment run, after the document stream was exhausted.

Currently, DRIFT does not maintain an inverted index—all searching is done using the query and document vectors directly. Because DRIFT needed to handle evaluation at intermediate points, rebuilding the inverted index at each evaluation point would have been necessary. Though considerable progress has been made in incremental updating [24–26], we wanted to concentrate on other issues.

When building the prototype DRIFT, we leveraged off of existing software as much as possible. For example, DRIFT has no stemming and stopwording capabilities—we used the unmodified SMART v11.0 software (available from Cornell at <ftp://ftp.cs.cornell.edu/pub/smart>) to do all stemming and stopwording and to produce simple term frequency (tf) document vectors. These vectors are then converted to a DRIFT format that is well suited for nonsequential access. These simple tf vectors, along with some auxiliary information kept in other files, form the source for the document stream.

## 5.2. Test Collections and Processing

In our experiments, we used four document collections as sources for the document stream. Two of these collections, MED and CACM, are well-known small collections.

TABLE 1  
Attributes of the Four Document Collections Used in the Experiments

Collection	Size	Num docs	Unique terms	Queries	Num rel docs
MED	1.1	1033	7170	30	696
CACM	2.2	3204	7153	64	796
AP89	266.5	84678	118242	50	2834
WSJ91	145.9	42652	72022	50	1109

The other two collections, AP89 and WSJ91, are subsets of the data used in the TREC conferences [23, 27, 28] and are much larger. For the smaller collections, we used all of the queries supplied. For the larger collections, we used only queries 151–200. The general attributes of the collections are listed in Table 1.

Since our goal was to determine the dissemination level of CWI for a particular IR model, we made no systematic attempt to determine the best combination of stoplist, stemming, and term weights for each document collection. However, guided by Salton and Buckley [29], we chose what we view as reasonable values for each and fixed them for the experiments reported here. For all collections, we used the stoplist and word stemming capabilities that come with the SMART v11.0 software. We only considered term weights that have a collection-wide component to them. In particular, this means the constituents of the idf: the total number of documents ( $N$ ) and the document frequency ( $df_k$ ) for each term. For all collections, we used cosine normalized (tfc) term weights for documents and unnormalized weights (nfx) for query term weights. The tfc and nfx notation is from Salton and Buckley [29]. The tfc weighting is given by

$$w_k = \frac{tf_k idf_k}{\sqrt{\sum_{i=1}^{n_{Terms}} (tf_i idf_i)^2}} \quad (4)$$

and the nfx weighting used for queries is given by

$$w_k = \left( 0.5 + 0.5 \left( \frac{tf_k}{\max tf} \right) \right) idf_k. \quad (5)$$

## 5.3. Experimental Runs

A single run (repetition) in our experiments involved fixing values for the various configuration parameters (see Table 2). The entire stream of documents taken from just one of the collections above was inserted into the distributed archive and effectiveness was measured at the end of the run using queries and relevance judgements associated with the source of the document stream. For each run, 11 point recall/precision numbers were recorded for each

TABLE 2  
Selected Parameters from  
the Configuration File for a  
Single Run

DissemLevel	0.5
AffinityLevel	1.0
RandomSeed	6582
NumberSites	20
CollectionName	med
DocWeight	tfc
QueryWeight	nfx

query. A user-level average was calculated from the results of all queries. In all of our runs, we compare the results against those obtained from the centralized collection made up of the combined holdings of the distributed collection.

To examine the effect of dissemination at various content-skew levels, we performed runs at  $a = 0.0$  and  $a = 1.0$  for 10 dissemination levels uniformly distributed between  $d = 0.0$  and  $d = 0.9$ . To examine the effect of content-skew, we fixed  $d$  at 0 and performed runs at 11 affinity levels between  $a = 0.0$  and  $a = 1.0$ .

There is a stochastic element to the allocation of documents to sites. With incomplete dissemination ( $d < 1$ ),  $G_i$  for any site  $i$  will differ from run to run. To allow for this variation, we performed 10 repetitions for each combination of configuration parameters (collection, dissemination, and affinity). In all of our figures we show the mean of these repetitions. In all configurations we fixed the number of sites at 20.

## 6. RESULTS

Figure 7 shows effectiveness for the four test collections with  $s = 20$ ,  $a = 0.0$ , and various levels of dissemination. For the two small collections, effectiveness was slightly reduced when there was no communication between sites ( $d = 0.0$ ). A small increase in dissemination from 0 to 0.2 boosted precision at all recall levels to be essentially indistinguishable from the central archive. For the two larger collections, effectiveness was comparable to the central archive regardless of the dissemination level.

In Fig. 8, we show results when  $s = 20$ , high affinity ( $a = 1.0$ ), and varying degrees of dissemination. For all collections, we see much larger differences in precision as dissemination changes than we did for low affinity. In all cases, effectiveness increases monotonically with increasing  $d$ . The level of dissemination at which effectiveness was comparable to the central archive was  $d = 0.5$  for the AP89 and WSJ91 collections,  $d = 0.6$  for CACM, and  $d = 0.8$  for MED. The greatest jump in effectiveness oc-

curred at low dissemination levels. Successive jumps in dissemination past the  $d = 0.1$  mark yielded relatively lower effectiveness gains.

We also were interested in how varying the content of subcollections affected retrieval. In Fig. 9, we show effectiveness for all four collections with  $s = 20$ ,  $d = 0.0$ , and varying levels of affinity. As in the results for varying dissemination, we see monotonic increases in precision as we change the parameter of interest. Unlike these previous results, the changes in effectiveness are much more linear: a change in affinity of  $\Delta a$  yields a corresponding change in precision of  $\Delta p$ .

Whenever  $s > 1$  and  $d < 1$  we can expect some variation in effectiveness due to the stochastic component of document allocation. We show typical variation in precision in Fig. 10 for the MED collection for selected values of  $a$ . The error bars represent  $\pm$  one standard deviation. Surprisingly, the variation is very small, regardless of  $a$ . We saw similar variation for all combinations of affinity, dissemination, and collections.

## 7. DISCUSSION

### 7.1. Collection Size

There is clear evidence of “start-up” behavior in our experiments. For small collections, the size of an individual site’s holdings,  $N_i$ , was very small. For example, the MED collection has 1033 documents, so  $N_i \approx 50$ . Even for nonskewed ( $a = 0.0$ ) collections, it is likely that a 50-document sample is not large enough to adequately characterize CWI. In systems with low dissemination, we would expect effectiveness to be degraded as long as CWI is not adequately representative of the entire collection. Our results support this expectation. For CACM and MED, we observed small effectiveness degradations (Fig. 7, top) at  $d = 0.0$  for random document allocations. Increases in dissemination to  $d = 0.2$  essentially raise effectiveness to that of a central archive. This boost in dissemination also increases the number of documents represented in CWI from  $N_i$  to  $N_i + d(\sum_{j=1, j \neq i}^s N_j)$  for site  $i$ . For MED and CACM, this is a jump from 50 to 240 documents and 160 to 768 documents respectively. For the larger collections,  $N_i$  is already very large, 4230 for AP89 and 2130 for WSJ91. In these cases, we did not observe effectiveness degradations when  $a = 0.0$  regardless of the dissemination levels (Fig. 7, bottom). In previous work [30] we hypothesized that some minimal sample of documents was needed to achieve search quality comparable to that of a central archive, but whether this sample was expressed as a fraction of the total archive size or as a minimal number of documents was unclear. The latter interpretation appears to be the appropriate one.

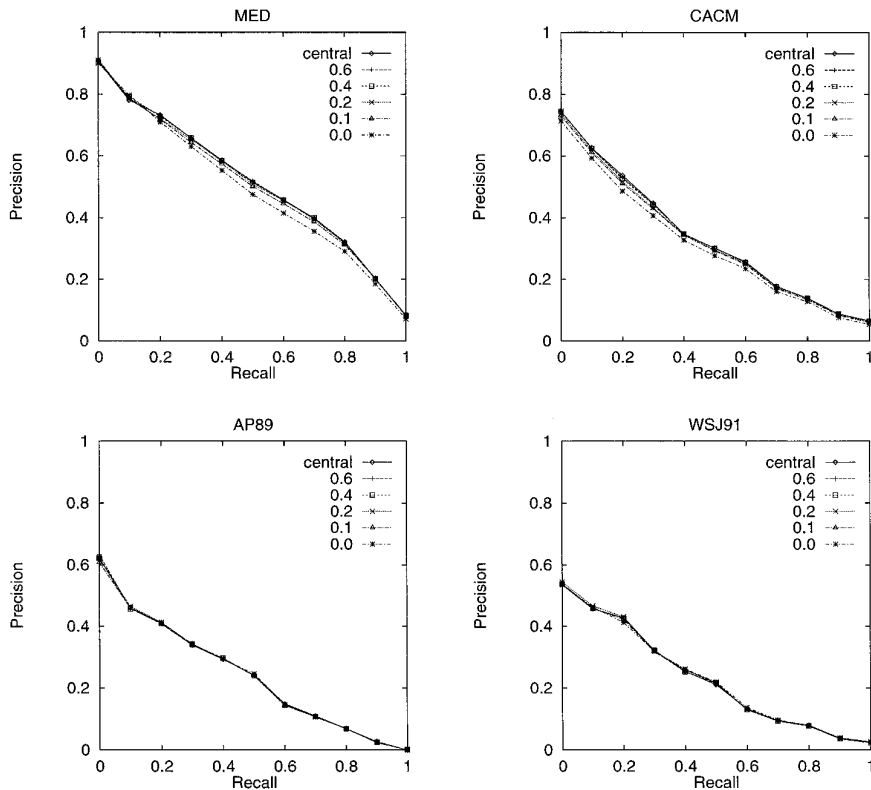


FIG. 7. Retrieval effectiveness on four test collections with 20 sites, no document clustering ( $a = 0.0$ ) and varying levels of dissemination. Note the high degree of overlap, particularly for AP89 and WSJ91.

In content-skewed collections (Fig. 8) we observed marked effectiveness degradations at low dissemination levels in all collections, but saw greater differences in the smaller collections. This is likely due to the dependence of our allocation model on relevance judgements. As a proportion of total collection size, the larger collections have relatively fewer relevant documents compared to the smaller collections. This means that the realized content-skew for the larger collections was somewhat less than for the smaller collections, and so we saw less effectiveness degradation.

## 7.2. Iso-knowledge and Iso-effectiveness

The dissemination model presented in Section 4.1 has some interesting properties. Using Equation (3) and knowledge of the size of the local collections, we can determine the total proportion of documents represented by  $C_i^g$ . Let this proportion be  $k_i$  and let  $c_i = N_i/N$  be the fraction of all documents held at site  $i$ . Then

$$k_i = c_i + d \left( \sum_{j=1, j \neq i}^s c_j \right), \quad \text{where } \sum_{j=1}^s c_j = 1. \quad (6)$$

When local collections are all the same size, then  $c_i = 1/s$  and we have a global  $k$  defined by

$$k = \frac{1}{s} + d \left( \frac{s-1}{s} \right). \quad (7)$$

In both equations, the first term represents the contribution of the local site and the second term the contribution of all the other sites. If we fix  $k$ , then we can generate *iso-knowledge* lines by varying  $s$  and solving for  $d$  or vice versa. Isoknowledge lines for three values of  $k$  are shown in Fig. 11.

Any point on an isoknowledge line represents a distributed archive configuration (an  $\langle s, d \rangle$  pair) that defines a system with knowledge about the distributed collection equivalent to a system configured from  $s$  and  $d$  chosen from any other point on the line. For example, point A in Fig. 11 represents a system of 10 sites disseminating at  $d = 0.111$ . Each site in this system has the same amount of knowledge about the global collection as the system represented at point B, a 20 site system disseminating at  $d = 0.158$ .

In earlier work [30], we presented evidence that these isoknowledge lines were also isoeffectiveness lines, since all distributed archive configurations ( $\langle s, d \rangle$  pairs) on a particular line had knowledge about the same number of documents and thus should perform similarly. Our recent work with the larger collections shows that though

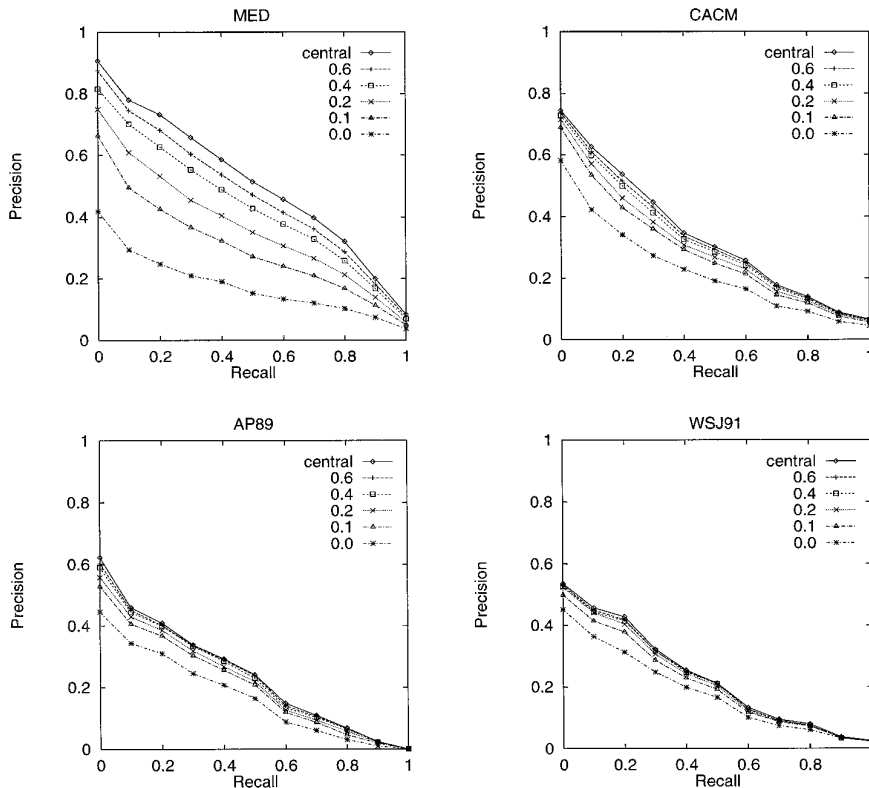


FIG. 8. Retrieval effectiveness on four test collections with 20 sites, maximal document clustering ( $a = 1.0$ ), and varying levels of dissemination.

this hypothesis is true, it is vacuous once a site has a reasonable sized sample of the entire distributed holdings. When this point is reached, systems perform well regardless of the  $k$  level. We make two additional points: (1) this new insight underscores the problems of working only with small datasets; and (2) differential effectiveness is still possible in operational scenarios where  $k \times N$  is small.

### 7.3. Pathological Cases

The normal function of the idf is to improve retrieval effectiveness by assigning high weights to those terms that are good discriminators, i.e., that appear in only a few documents. When similar documents are clustered together at the same site and dissemination is incomplete (or nonexistent), then idf weighting can have exactly the opposite effect. Terms appearing rarely in the global collection may appear often in the local collection, causing the corresponding term weights to be low. Figure 12 illustrates this phenomenon. In the MED collection, term frequency alone achieves better effectiveness than when the idf is included and dissemination is low. While such behavior is not guaranteed when similar documents are collocated, it is clearly possible. This may appear to be an argument for

term frequency weighting, but we also note that a relatively small amount of dissemination ( $d = 0.4$  in this case) permits superior performance for the idf-based term weighting scheme.

### 7.4. Implications and Scalability

The relatively low amounts of dissemination needed to maintain retrieval effectiveness have some interesting implications. In dynamic applications such as filtering and routing [31, 32], completely up-to-date CWI may not be needed, so recalculation of CWI need be done only intermittently. Many resource discovery systems (e.g., Callan *et al.* [13] or Gravano *et al.* [33]) use CWI to select a small number of collections to send queries to. Our work indicates that this CWI may drift considerably without unduly harming search quality.

Without additional machinery to prune the number of sites involved in a search, the distributed architecture presented here will not scale to a very large number of sites. For this reason we did not present results for large  $s$ , nor do we imagine that this architecture will be used for large  $s$  without some of the machinery just mentioned. However, it is easy to imagine a smaller system (tens of sites) where such an architecture is practical. Our results show that

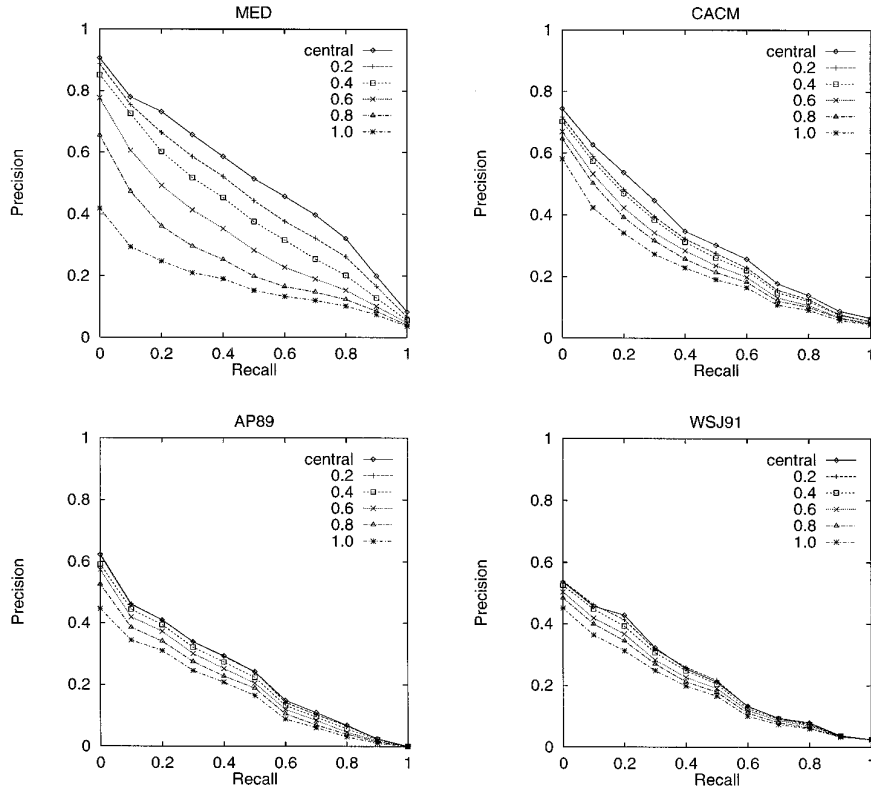


FIG. 9. Retrieval effectiveness on four test collections with 20 sites, no dissemination of collection-wide information ( $d = 0.0$ ), and varying levels of document clustering.

even for such relatively small-scale systems, dissemination of CWI is needed.

## 8. CONCLUSIONS

The dissemination model presented here has intuitive appeal. The two extremes of the model describe a distrib-

uted archive with either no communication or complete communication between sites.

Our experiments show that even for modest-sized distributed archives (20 sites), dissemination of CWI is needed to maintain retrieval effectiveness. Surprisingly, complete dissemination is not required to achieve good effectiveness.

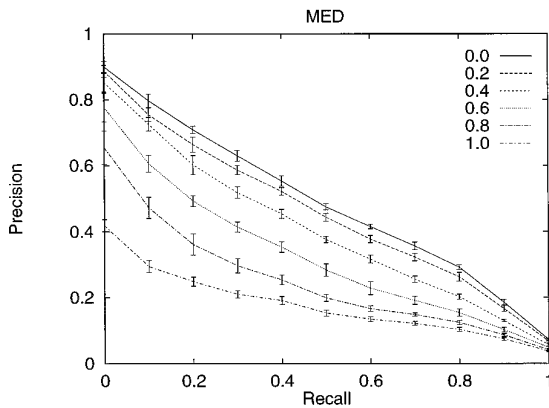


FIG. 10. Variation in retrieval effectiveness at affinity levels from 0.0 to 1.0. The error bars at each recall level represent  $\pm$  one standard deviation (10 runs). This is the MED collection with 20 sites.

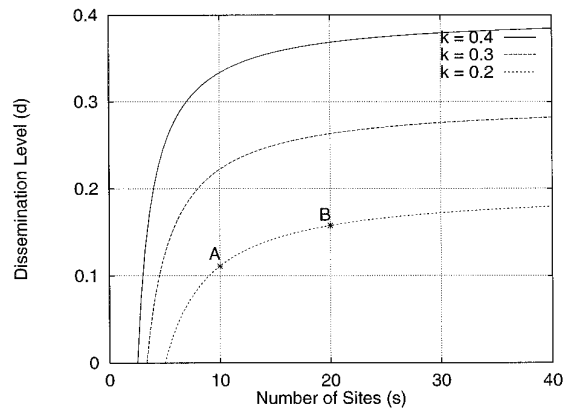


FIG. 11. Isoknowledge lines for three values of  $k$ . The points marked A and B represent two combinations of  $s$  and  $d$  that define systems with the same amount of knowledge about the global document collection.

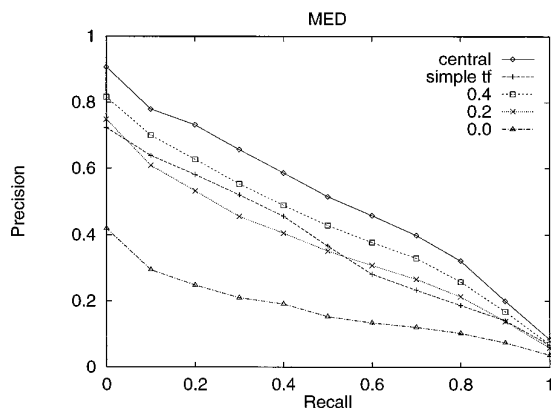


FIG. 12. When dissemination is imperfect ( $d < 1$ ) and documents are clustered by their content, then effectiveness can be greatly reduced. We show the MED collection with 20 sites,  $a = 1.0$ , idf-based term weighting and various dissemination levels. The dotted line with “+” shows effectiveness on the same collection for simple term frequency weighting.

More dissemination is needed when documents are allocated to sites based on their content than when they are randomly allocated.

The examination of both small and large test collections indicates that operational systems with low levels of intersite communication may well exhibit poorer retrieval quality initially, but show better effectiveness as the underlying archive grows.

## 9. SUMMARY AND FUTURE WORK

Distributed information retrieval systems will be the bedrock technology upon which successful digital library systems will be constructed. The research reported here has helped us to understand the effects on the information retrieval model and the influences on retrieval performance implied by the distribution of information across the multiple sites of a distributed information retrieval system. It extends the earlier work of Viles and French to larger, more realistic data sets and corroborates their findings [30].

A related study on the effects of update activity (dynamism) on IR systems was reported in Viles and French [34]. The findings in that study are wholly consistent with those reported here, but derived solely from the empirical study of small test collections. We are now extending that work to larger data sets and will report those results in a forthcoming paper.

Finally, both the studies above [30, 34] show conclusively that the topical distribution of the collection content can have a direct influence on the performance of the resulting system. This phenomenon, which we have called *content skew*, is captured by the interplay between the dissemina-

tion parameter,  $d$ , and the affinity parameter,  $a$ , defined in the studies. We are now in the process of analyzing the content skew of the recently deployed NCSTRL system. This analysis is interesting in its own right and will help us to access the effect of content skew in an operational system.

Our approach to the analysis of content skew uses cluster techniques to map the topical content of all the sites participating in NCSTRL. We are using a two step process:

1. intrasite document clustering to establish the topical coverage at each site, and
2. intersite topic clustering to infer topic distribution across the sites.

This work is described more fully in a forthcoming paper.

We expect the work reported here to give us some insight into aspects of operational digital libraries that will help guide engineering decisions that affect the performance of such systems. A recent workshop on research issues in digital libraries [2] concluded that the only way to effectively examine issues in digital libraries is to build and deploy them as operational systems. The work described here embraces that concept fully.

## ACKNOWLEDGMENTS

This work was supported in part by NASA Goddard Space Flight Center under GSRP Fellowship NGT-51018 and by NASA CESDIS Grant 5555-25. We also thank Allison Powell for help in preparing some of the figures and the referees for their careful reading of the paper.

## REFERENCES

1. H. M. Gladney, N. J. Belkin, Z. Ahmed, E. A. Fox, R. Ashany, and M. Zemankova, *Digital Library: Gross Structure and Requirements (Report from a Workshop)*, Technical Report RJ-9840, IBM, May 1995.
2. C. Lynch and H. Garcia-Molina (Eds.), *Report on the May 18–19, 1995 IITA Digital Libraries Workshop, 1995*. <http://www.diglib.stanford.edu/diglib/pub/reports/iita-dlw/main.html>.
3. Marc VanHeyningen, The Unified Computer Science Technical Report Index: Lessons in indexing diverse resources, in *2nd International World Wide Web Conference, WWW'94, Chicago, October 1994*, pp. 535–543.
4. C. Mic Bowman, Peter B. Danzig, Darren Hardy, Udi Manber, and Michael F. Schwartz, The Harvest Information Discovery and Access System, in *2nd International World Wide Web Conference, WWW'94, Chicago, October 1994*, pp. 763–771.
5. James R. Davis and Carl Lagoze, Drop-in publishing with the World-Wide Web, in *2nd International World Wide Web Conference, WWW'94, Chicago, October 1994*, pp. 749–758.
6. Carl Lagoze and James R. Davis, Dienst: An architecture for distributed document libraries, *Comm. ACM* **38**(4), 1995, 47.
7. James C. French, Edward A. Fox, Kurt Maly, and Alan Selman, Wide Area Technical Report Service: Technical reports online, *Comm. ACM* **38**(4), 1995, 45.
8. James C. French, Edward A. Fox, Kurt Maly, and Alan Selman,

- Wide Area Technical Report Service, in *2nd International World Wide Web Conference, WWW'94, Chicago, October 1994*, pp. 523–533.
9. J. C. French, *Electronic Distribution of Technical Reports and Working Papers: A Simple Cooperative Approach*, Technical Report CS-92-27, University of Virginia, 1992.
  10. Edward A. Fox, World-Wide-Web and computer science technical reports, *Comm. ACM* **38**(4), 1995, 43–44.
  11. J. R. Davis and C. Lagoze, *A Protocol and Server for a Distributed Digital Technical Report Library*, Technical Report TR94-1418, Cornell University, 1994.
  12. C. Lagoze, E. Shaw, J. R. Davis, and D. B. Krafft, *Dienst: Implementation Reference Manual*, Technical Report TR95-1514, Cornell University, 1995.
  13. James P. Callan, Zhihong Lu, and W. Bruce Croft, Searching distributed collections with inference networks, in *Proceedings of the 18th International Conference on Research and Development in Information Retrieval (SIGIR95), Seattle, 1995*, pp. 21–29.
  14. C. Mic Bowman, Peter B. Danzig, Udi Manber, and Michael F. Schwartz, Scalable Internet resource discovery: Research problems and approaches, *Comm. AMC* **37**(8), 1994, 98–107.
  15. I. J. Aalbersberg and Frans Sijstermans, High-quality and high-performance full-text document retrieval: The Parallel InfoGuide system, in *Proc. First Intl. Conf. on Parallel and Distributed Information Systems, Miami Beach, FL, 1991*, pp. 142–150.
  16. Gerard Salton and Michael McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.
  17. Charles L. Viles, Maintaining state in a distributed information retrieval system, in *32nd Annual ACM Southeast Regional Conference, Tuscaloosa, AL, March 17–18, 1994*, pp. 157–161.
  18. Zygmunt Mazur, On a model of distributed information retrieval systems based on thesauri, *Inform. Process. Management* **20**(4), 1984, 499–505.
  19. Donna Harman, Wayne McCoy, Robert Toense, and Gerald Candela, Prototyping a distributed information retrieval system using statistical ranking, *Inform. Process. Management* **27**(5), (1991), 449–460.
  20. Ellen Vorhees, Narendra K. Gupta, and Ben Johnson-Laird, Learning collection fusion strategies, in *Proceedings of the 18th International Conference on Research and Development in Information Retrieval (SIGIR95), Seattle, Wash., 1995*, pp. 172–179.
  21. Nicholas J. Belkin, Paul Kantor, Edward A. Fox, and J. A. Shaw, Combining the evidence of multiple query representations for information retrieval, *Inform. Process. Management* **31**(4), 1995, 431–448.
  22. Anthony Tomasic and Hector Garcia-Molina, Query processing and inverted indices in shared-nothing text document information retrieval systems, *VLDB J.* **2**(3), 1993, 243–275.
  23. Donna Harman, Overview of the First Text Retrieval Conference (TREC-1), in *Proceedings of the First Text Retrieval Conference (TREC-1), Gaithersburg, MD, 1992*, pp. 1–20.
  24. Eric W. Brown, James P. Callan, and W. Bruce Croft, Fast incremental indexing for full-text information retrieval, in *Proceedings, 20th VLDB Conference, Santiago, Chile, 1994*, pp. 192–202.
  25. Anthony Tomasic, Hector Garcia-Molina, and Kurt Shoens, Incremental updates of inverted lists for text document retrieval, in *SIGMOD94, Minneapolis, May 1994*, pp. 289–300.
  26. Justin Zobel, Alistair Moffat, and Ron Sacks-Davis, An efficient indexing technique for full-text database systems, in *Proceedings, 18th VLDB Conference, Vancouver, 1992*, pp. 352–362.
  27. Donna Harman. Overview of the Second Text Retrieval Conference (TREC-2), in *Proceedings of the Second Text Retrieval Conference (TREC-2), Gaithersburg, MD, 1993*, pp. 1–20.
  28. Donna Harman. Overview of the Third Text Retrieval Conference (TREC-3), in *Proceedings of the Third Text Retrieval Conference (TREC-3), Gaithersburg, MD, 1994*, pp. 1–19.
  29. Gerard Salton and Christopher Buckley, Term-weighting approaches in automatic text retrieval, *Inform. Process. Management* **24**(5), 1988, 513–523.
  30. Charles L. Viles and James C. French, Dissemination of collection wide information in a distributed information retrieval system, in *Proceedings of the 18th International Conference on Research and Development in Information Retrieval (SIGIR95), Seattle, July 1995*, pp. 12–20.
  31. James C. French, DIRE: An approach to improving scientific communication, *Inform. Decision Technol.* **19**, 1994, 527–541.
  32. Shoshana Loeb and Douglas Terry, Information filtering, *Comm. ACM*, **35**(12), 1992, 26–28.
  33. Luis Gravano, Hector Garcia-Molina, and Anthony Tomasic, *The Efficacy of GLOSS for the Text Database Discovery Problem*, Technical Report STAN-CS-TN-93-2, Stanford University, 1993.
  34. Charles L. Viles and James C. French, On the update of term weights in dynamic information retrieval systems, in *Proceedings of the 4th International Conference on Knowledge and Information Management, Baltimore, MD, November 1995*, pp. 167–174.



JAMES C. FRENCH received a B.A. in mathematics and M.S. and Ph.D. (1982) degrees in computer science, all at the University of Virginia. After several years in industry, he returned to the University of Virginia in 1987 as Senior Scientist in the Institute for Parallel Computation and became a research assistant professor in 1990. His current research interests include information retrieval in widely distributed systems. Professor French is a member of the ACM, the IEEE Computer Society, SIAM, and Sigma Xi.



CHARLES L. VILES received a B.S. in forestry (1984) from Virginia Tech and an M.S. in computer science (1988) from William and Mary. After his M.S., he worked for three years at the Virginia Institute of Marine Science building software and hardware systems for digital image analysis of oceanic plankton. In 1991, he entered the University of Virginia, where he is currently a Ph.D. candidate in computer science. His professional interests include information retrieval, wide area networks, and computer science education.