Dissemination of Collection Wide Information in a Distributed Information Retrieval System *

Charles L. Viles and James C. French

Department of Computer Science
University of Virginia
Charlottesville, VA 22903

January 6, 1995

Technical Report CS-95-02 (*Submitted to SIGIR95*)

Abstract

We find that dissemination of collection wide information (CWI) in a distributed collection of documents is needed to achieve retrieval effectiveness comparable to a centralized collection. Complete dissemination is unnecessary. The required dissemination level depends upon how documents are allocated among sites. Low dissemination is needed for random document allocation, but higher levels are needed when documents are allocated based on content. We define parameters to control dissemination and document allocation and present results from four test collections. We define the notion of iso-knowledge lines with respect to the number of sites and level of dissemination in the distributed archive, and show empirically that iso-knowledge lines are also iso-effectiveness lines when documents are randomly allocated.

1 Introduction

In the rapidly evolving internetworks of today, we see a vast diversity of information becoming electronically available. The information environment is highly distributed, highly dynamic, and extremely heterogeneous. One of the great research challenges posed by this environment is the efficient and effective search of the vast distributed archive for useful information. Though work in this arena is preliminary, most approaches rely on maintenance of a centralized archive - possibly

*This work was supported by NASA Goddard Space Flight Center under GSRP fellowship NGT-51018 and by NASA/CESDIS Grant 5555-25.

replicated, possibly with caches - built by navigating the Internet and down-loading documents to a central repository [2, 7, 13, 15]. Efficiency is the overriding concern. In this paper, we are concerned with retrieval effectiveness. How can we achieve effectiveness comparable to 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 maintain retrieval effectiveness commensurate with a central archive. Our contributions include:

- A model for CWI dissemination within a distributed collection of documents.
- The definition of *iso-knowledge* and *iso-effectiveness* lines and a demonstration of their utility.
- A finding that relatively low dissemination is needed for distributed collections where documents are randomly allocated to sites.
- A finding that higher dissemination levels are needed when documents are allocated to sites so that similar documents are more likely to be co-located.

We start by presenting related work in Section 2. In Section 3, we describe the distributed archive and provide a description of CWI dissemination, document allocation, and the parameters we use to model these attributes. We continue in Section 4 with a description of our IR software, followed by a description of our experiments in Section 5. Section 6 details our results. A discussion of some of the issues and questions raised by our work is in Section 7. We finish with a summary and some directions for future work.

2 Related Work

Bowman *et al.* [3] describe many of the issues involved in resource discovery and information retrieval on the Internet. The Harvest [2] system is a prototype resource discovery and access system designed to address some of these problems. 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 [1], Aalbersberg and Sijstermans use a distributed-memory multi-processor to get very fast query response times. They use the Vector Space Model [16, 17] 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 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 [21] 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 [14] 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.* [10] 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.

Several systems have been built to make computer science technical reports available. UCSTRI [20] operates by proactively fetching summary files from many sites and indexing them centrally. WATERS [12] maintains a central index for searching and offers a distributed browse capability. Participating sites periodically send their bibliographic files to the central server for indexing. The CSTR project and its accompanying protocol, server, and bibliographic format [5, 6] provides search and browse access to the technical reports of participating members. The system supports

distributed searching and multiple formats for documents.

Work by Tomasic, Garcia-Molina and others [18, 19] has focused on performance issues e.g. distributed index architectures that provide low query response time. Effectiveness is not considered.

3 The Distributed Archive Model

In a distributed archive (Figure 1), 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.

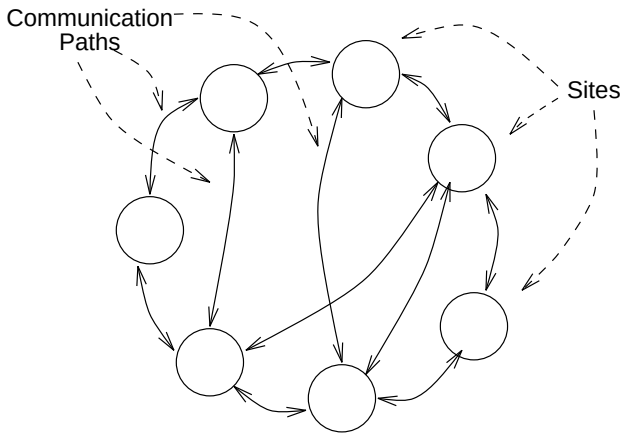

Figure 1: The topology for an instance of a distributed archive. Document insertions happen asynchronously at each site. Sites communicate with each other to exchange information on their respective collections and to answer queries.

Documents arrive in the system and are allocated to sites based upon some criteria. This criteria may be administrative, e.g. the document was created at site i so it resides at site i, or it may be content-based, e.g. the document is similar to these others so it will be co-located 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 judgements.

Examples of a distributed archive might include a distributed technical report archive, or doc-

umentation for a large distributed project like NASA's Mission to Planet Earth.

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.

3.1 Models for Dissemination and Allocation

The distributed archive is composed of s sites. The j-th document at site i is denoted by D_{ij}. At any site, 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^g represents the collection of documents that has been used to generate site i's version of CWI. We call this version G_i, so $G_i = f(C_i^g)$.

3.2 Dissemination of Collection Wide Information

3.2.1 Description

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. The *idf* information is used in many term-weighting schemes in a large number of IR models. In the TREC-1 [9] 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

$$idf_k = \log\left(\frac{N}{df_k}\right) \tag{1}$$

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

$$idf_k = \log\left(\frac{\sum_{i=1}^{s} N_i}{\sum_{i=1}^{s} df_{ik}}\right) \tag{2}$$

where N_i and 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.[1] Given distributed search, each site must know the global idf for a faithful implementation of the operative IR model.

The addition of a single document causally effects the CWI. In a completely faithful implementation of an IR model using CWI, this would require dissemination of 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?
- How should CWI by circulated?

We consider only the first question in this paper, as it has a profound influence on the answer to the second question. Check [21] for an algorithm that addresses the second question.

3.2.2 Dissemination Model

We model dissemination of CWI as follows. Let $prefix(d, C_i^l)$ 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 $prefix(d, C_j^l)\ \forall j \neq i$. That is

$$C_i^g = C_i^l \cup \left(\bigcup_{j \neq i} prefix(d, C_j^l) \right) \tag{3}$$

We note the following about d, 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.

[1] Sites must agree on the identity of the k-th term.

- When $d = 1$, complete dissemination occurs. Every site has "perfect" knowledge of every other site. Local estimates of CWI are identical to each other and CWI derived from the union of all local collections.

Figure 2 illustrates this dissemination for $d = 0.25$.

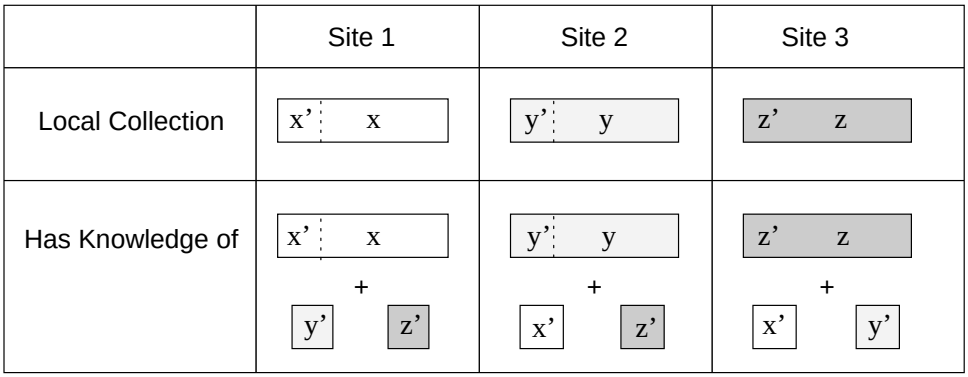

Figure 2: The degree of dissemination found in a distributed archive with 3 sites and $d = 0.25$. The horizontal blocks represent the stream of documents stored at each site, in the order they were inserted. Each site knows about its own documents and the first 25% of the documents inserted at other sites.

3.3 Allocation

3.3.1 Description

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

3.3.2 Allocation Model

Qualitatively, we wish to see how varying the allocation of documents to sites affects retrieval performance. The criterion we define here is convenient for experimentation but should not be construed as a recommendation for document clustering. [2]

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 pieces of information:

- relevance information,
- $QHome()$,
- 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: $a + (1-a)\frac{1}{s}$ 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 documents are not relevant to more than one query: not completely realistic, but reasonable to a first approximation for the collections we used in our tests. Figure 3 shows the algorithm in pseudo-code, and Figure 4 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 two affinities.

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 the content of a document has nothing to do with its location.
- When $a = 1$, documents relevant to the same query are colocated, mapping to the case where document content has a large influence on document location.

Our goal in defining affinity in this manner is not to achieve the best possible "clustering" of documents. It is clear that documents not relevant to any query are randomly allocated regardless

[2] In fact, in an operational environment, our criterion is problematic because it requires known queries and relevance judgements.

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

Figure 3: 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 $1 \leq j \leq$ numSites.

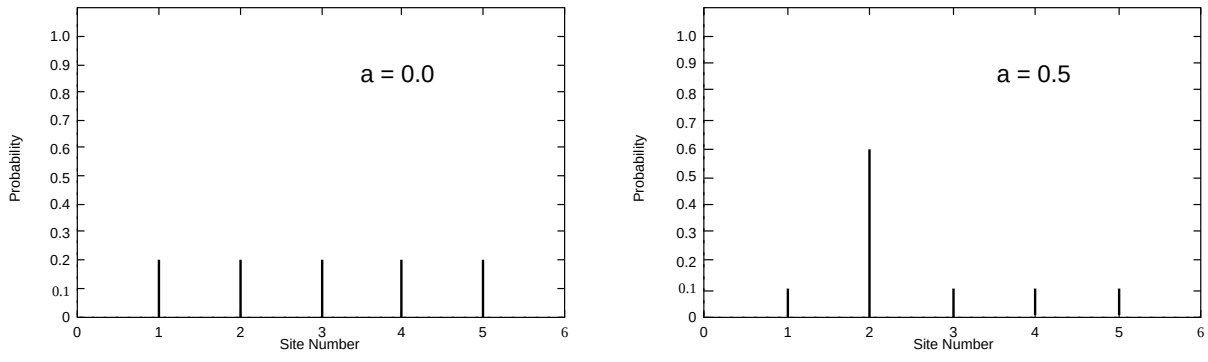

Figure 4: Probability that a document D will be assigned to a site given 5 sites, D is relevant to Q, $QHome(Q) = 2$, for two levels of affinity.

of *a* or their content. However, the clustering does allow us to determine the level of dissemination of CWI needed for various allocations of documents, and for this reason is useful.

4 Description of Software

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, 17] 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 collection wide information. 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 can also 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 the document stream. 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 [4, 19, 22], we wanted to concentrate

on other issues. We anticipate re-examining the indexing question when we start to look at larger collections.

When building the prototype DRIFT, we leveraged off of existing software as much as possible. For example, DRIFT has no stemming and stoplisting 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 stoplisting and to produce simple term frequency (tf) document vectors. [3] These vectors are then converted to a DRIFT format that is well-suited for non-sequential access. These simple tf vectors, along with some auxiliary information kept in other files, form the document source for the *DocStream* object.

5 Description of Experiments

5.1 Test Collections and Processing

In our experiments, we used four document collections, CRAN, CISI, CACM, and MED as the source for the document stream. All of these collections are available via anonymous ftp from *ftp://ftp.cs.cornell.edu/pub/smart*. The attributes of these collections are well-known and are not repeated here.

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, term weights, and similarity functions for each document collection. However, 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 comes 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. Emulating the SMART configuration files for the four test collections, we used un-normalized term weights for CACM and CRAN:

$$w_k = 0.5 + 0.5 \left(\frac{tf_k}{\max tf} \right) idf_k \tag{4}$$

and for MED and CISI we used normalized term weights.

[3] Much of our configuration philosophy has been inspired by the SMART software as well.

$$w_k = \frac{0.5 + 0.5(\frac{tf}{\max tf})idf_k}{\sum_{i=1}^{nTerms}(0.5 + 0.5(\frac{tf_i}{\max tf})idf_i)} \quad (5)$$

A single run (repetition) in our experiments involved fixing values for the various configuration parameters (see Table 1). 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 query. A user-level average was calculated from the results of all queries.

DissemLevel	0.5
AffinityLevel	1.0
RandomSeed	6582
NumberSites	20
CollectionName	med
SimFunction	cosine

Table 1: Selected parameters from the configuration file for a single run.

There is a stochastic element to the allocation of documents to sites. When there is incomplete dissemination ($d < 1$), G_i for any site 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 allocation). In all of our figures we show the average of the 10 repetitions.

6 Results

Figure 5 shows effectiveness for the four test collections with $s = 20$, $a = 0.0$, and various levels of dissemination. In all four collections, effectiveness is slightly reduced when there is no communication between sites ($d = 0.0$), though the degree of reduction varies with the collection. For all collections, a small increase in dissemination from 0 to 0.2 boosted precision at all recall levels to be essentially indistinguishable from the central archive.

In Figure 6, 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

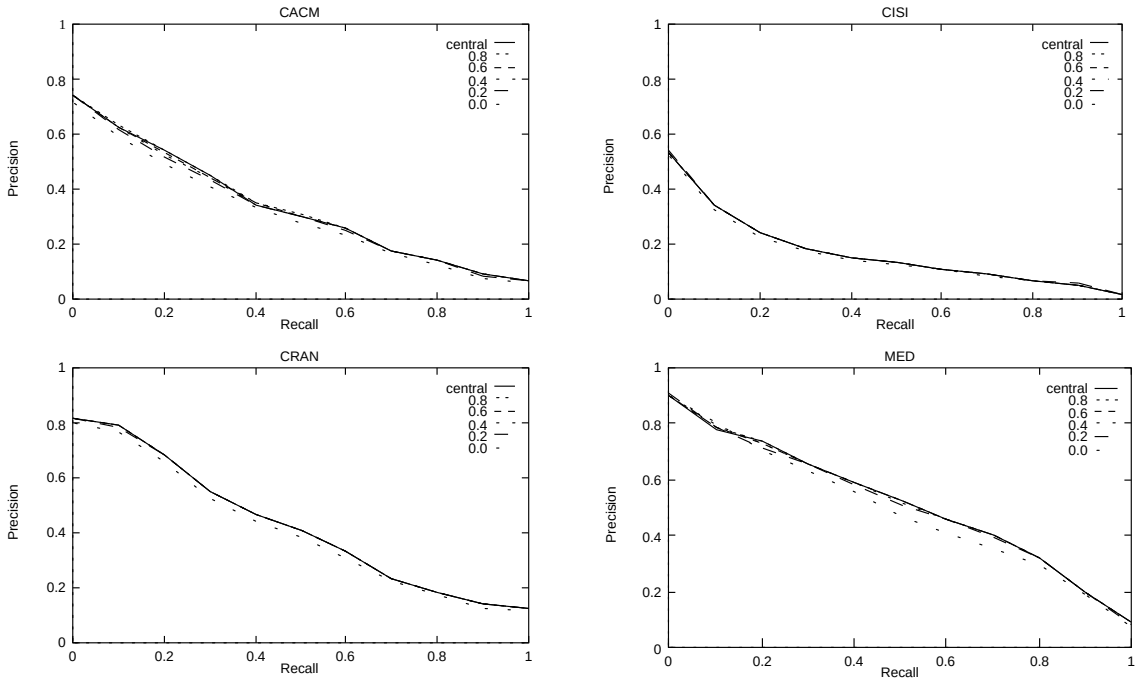

Figure 5: Retrieval effectiveness on four test collections with 20 sites, no document clustering ($a = 0.0$) and varying levels of dissemination.

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 $d = 0.4$ for the CRAN and CISI collections, $d = 0.6$ for CACM, and $d = 0.8$ for MED. As in the results for low affinity, the greatest jump in effectiveness occurs at low dissemination levels. Successive jumps in dissemination past the $d = 0.2$ mark yield relatively lower effectiveness gains.

We were also interested in how varying the clustering of documents affected retrieval. In Figure 7, 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 Δa yields a corresponding change in precision of Δp. Though not presented here, tests with $d = 0.5$ and varying affinity showed similar behavior, but the magnitude of Δp was smaller.

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 Figure 8 for the MED

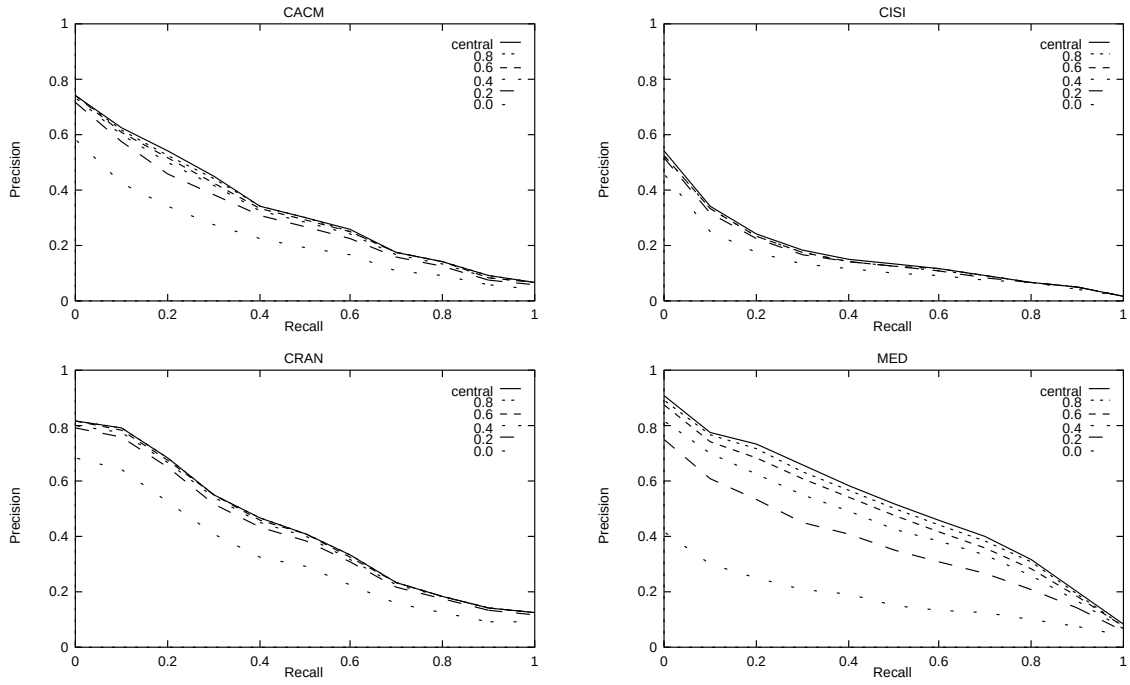

Figure 6: Retrieval effectiveness on four test collections with 20 sites, maximal document clustering ($a = 1.0$) and varying levels of dissemination.

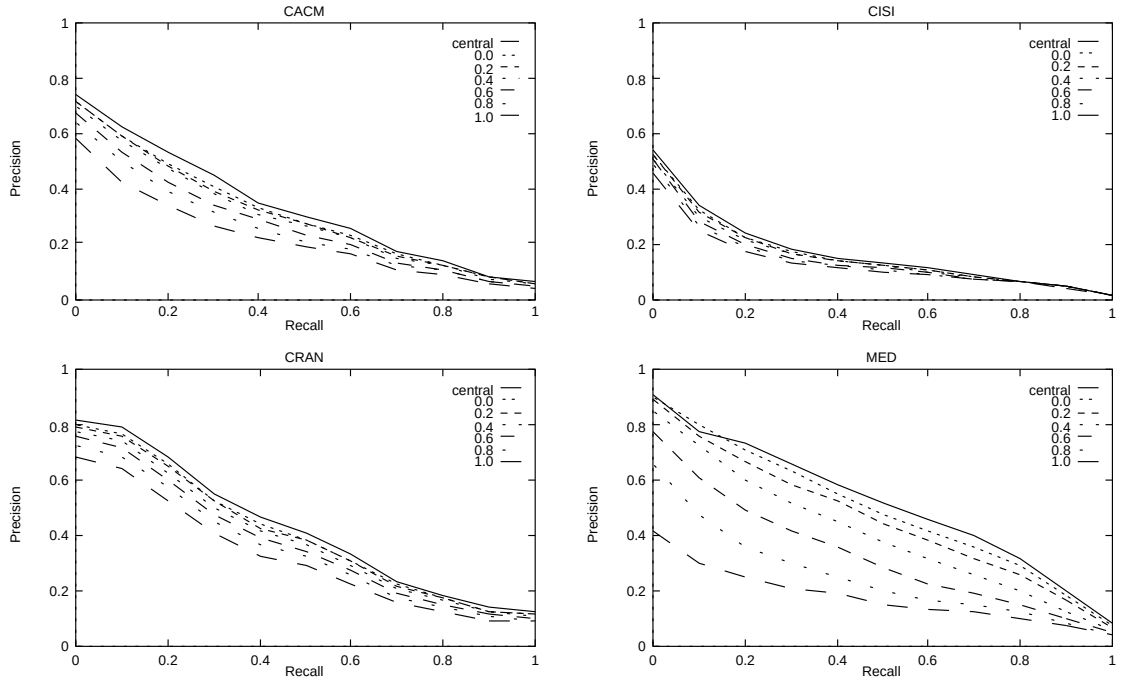

Figure 7: Retrieval effectiveness on four test collections with 20 sites, no dissemination of collection-wide information ($d = 0.0$) and varying levels of document clustering.

collection for selected values of a. The error bars represent $\pm$ one standard deviation. Surprisingly, the variation is very small, regardless of the a level. We saw similar or smaller variation for all combinations of affinity, dissemination, and collections.

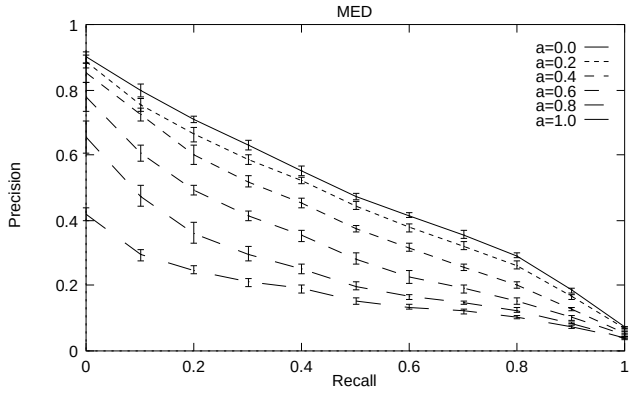

Figure 8: Variation in retrieval effectiveness at selected affinity levels. The error bars at each recall level represent $\pm$ one standard deviation (10 runs). This is the MED collection with 20 sites.

7 Discussion

The dissemination model presented in Section 3.2.2 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 $c_i = \frac{N_i}{N}$ be the fraction of all documents held at site i. Then

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

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

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

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. Iso-knowledge lines for three values of k are depicted in Figure 9.

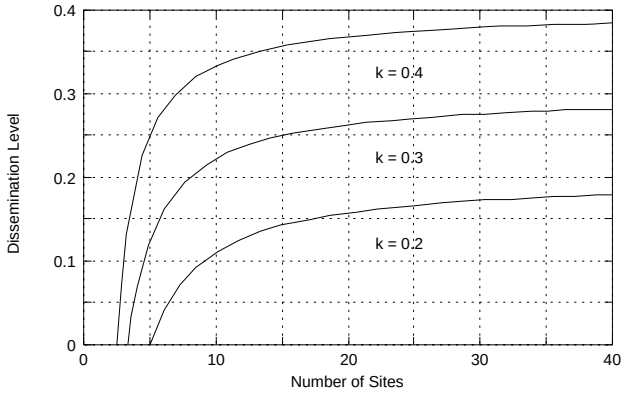

Figure 9: Iso-knowledge lines for three values of k. When documents are randomly allocated among all sites ($a = 0$), these lines are also iso-effectiveness lines.

If we assume that documents are randomly allocated to all sites ($a = 0$), then these iso-knowledge lines are also *iso-effectiveness* lines. Arguing informally, when random allocation is used, each site has a random sample of size ks from which G_i is calculated. The sample is unbiased because documents are placed without regard to their content or the locations of any other documents. On average, any sample of size ks will show effectiveness similar to any other sample of similar size. Empirical verification of this argument is presented in Figure 10 using (s, d) pairs from the iso-knowledge lines. Here, we show the relative difference in precision at the 11 recall levels for two pairs from the $k = 0.2$ iso-knowledge lines: $(5, 0.0)$ and $(20, 0.158)$. In all cases, the difference in precision is less than 5%, where the difference at recall level l is calculated as $100(\frac{precision_l^1 - precision_l^2}{precision_l^1})$.

The iso-effectiveness lines show that many combinations of dissemination level and number of sites achieve similar effectiveness. As a practical matter, the number of sites in the distributed archive is not a controllable parameter, so effectiveness must be maintained by controlling dissemination. The iso-effectiveness notion is important because it gives some direction for system designers and administrators. For example, if we use Equation 7, then a system with five sites and no dissemination must disseminate at $d = 0.111$ if five more sites are added and the same level of effectiveness is desired.

A consistent result across collections was that an increase in the dissemination level from 0 to 0.2 caused effectiveness to approach that of the centralized archive, at least when no clustering of documents was done. There appears to be some minimal sample of documents that a site needs to know about to achieve search effectiveness comparable to a central archive. It remains to be

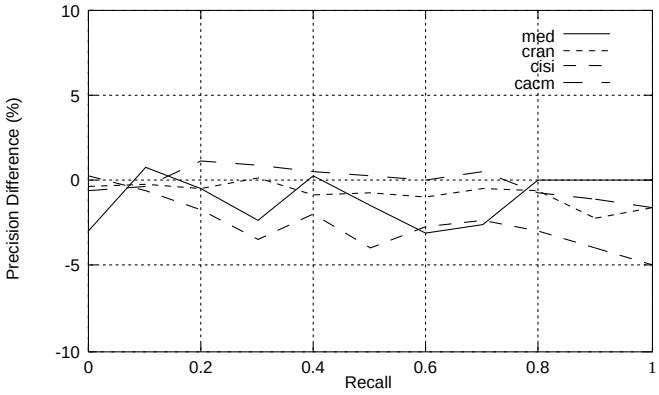

Figure 10: Empirical evidence of iso-effectiveness lines when documents are randomly allocated to each site. The y-axis shows the difference in precision between two (s, d) pairs chosen from the 0.2 iso-knowledge curve. The two pairs were (5, 0.0) and (20, 0.158). Four different collections are shown. Differences obtained from the average of 5 runs.

seen whether this sample is a fraction of the whole (as is suggested by the iso-knowledge curves of Figure 9), or some minimal number of documents is needed. Experiments with larger collections would help answer this question.

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 non-existent), 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 11 illustrates this phenomenon. Here, term frequency alone achieves better effectiveness than when the *idf* is included on the MED collection. While such behavior is not guaranteed when similar documents are co-located, 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) enables superior performance for the *idf*-based term weighting scheme.

The relatively low amounts of dissemination needed to maintain retrieval effectiveness has some interesting implications. In dynamic applications like filtering and routing [8, 11], completely up-to-date collection wide information may not be needed, so re-calculation of CWI need be done only intermittently.

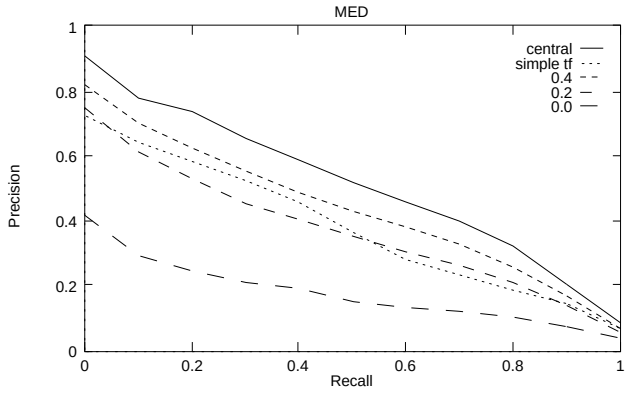

Figure 11: When dissemination is imperfect ($d < 1$) and documents are clustered by their content, then effectiveness can be greatly reduced. The solid and dashed lines show the MED collection with 20 sites, $a = 1.0$, *idf*-based term weighting and various dissemination levels. The dotted line shows effectiveness on the same collection for simple term frequency weighting.

The size of a collection can influence overall retrieval. Since we are working with fixed size test collections, varying the number of sites also changes the average collection size at any site. Because both collection size and number of sites have an effect on retrieval, it was not possible to attribute observed changes in effectiveness when the number of sites was varied solely to that parameter. For this reason, we elected to fix s for all of the work presented here. Additional experiments with larger collections might help us better assess this behavior.

The affinity parameter defined and used here was useful because it is a straightforward method to achieve non-uniform document allocation to sites. There are certainly more sophisticated methods to accomplish content-based document clustering.

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 just mentioned machinery. However, it is easy to imagine a smaller system (10's of sites) where such an architecture is practical. Our results show that even for such relatively small-scale systems, dissemination of CWI is needed.

8 Summary and Future Directions

The dissemination model presented here has intuitive appeal. The two extremes of the model describe a distributed archive with either no communication or complete communication between sites. Iso-knowledge and iso-effectiveness lines derived from the model indicate that as the number of sites increases, dissemination must increase to maintain a given level of knowledge and effectiveness. Empirical results support this analysis.

Our experiments show that even for modestly sized distributed archives (20 sites), dissemination of CWI is needed to maintain retrieval effectiveness. Surprisingly, complete dissemination is not required to achieve good effectiveness. More dissemination is needed when documents are allocated to sites based on their content than when they are randomly allocated.

We see several directions for future work. Tests with larger collections are needed to see whether trends reported here are persistent. Larger collections would also allow us to run experiments with larger numbers of sites. Further work examining iso-effectiveness for non-random document allocation is also needed.

References

[1] I. J. Aalbersberg and Frans Sijstermans. High-quality and High-performance Full-text Document Retrieval: The Parallel InfoGuide System. In *Proc. 1st Intl. Conf. Parallel and Distributed Information Systems*, Miami Beach, FL, 1991.

[2] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz. Harvest: A Scalable, Customizable Discovery and Access System. Technical Report CU-CS-732-94, Department of Computer Science, University of Colorado, August 1994.

[3] C. Mic Bowman, Peter B. Danzig, Udi Manber, and Michael F. Schwartz. Scalable internet resource discovery: Research problems and approaches. *Communications of the ACM*, 37(8):98–107, August 1994.

[4] Eric W. Brown, James P. Callan, and W. Bruce Croft. Fast Incremental Indexing for Full-Text Information Retrieval . In *Proc. 20th Conf. Very Large Databases*, Santiago, Chile, 1992.

[5] D. Cohen. A format for E-mailing Bibliographic Records. Technical Report Internet RFC 1357, Internet Engineering Task Force, July 1992.

[6] James R. Davis and Carl Lagoze. A protocol and server for a distributed digital technical report library. Technical Report TR-1418, Cornell University, April 1994.

[7] D. Eichmann, T. McGregor, and D. Danley. The RBSE Spider - Balancing Effective Search Against Web Load. In *Proc. 1st Intl. World Wide Web Conf.*, Geneva, Switzerland, May 1994.

[8] James C. French. DIRE: An Approach to Improving Scientific Communication. *Information and Decision Technologies*, 19:527–541, 1994.

[9] Donna Harman. Overview of the First Text Retrieval Conference (TREC-1). In *Proc. 1st Text Retrieval Conference (TREC-1)*, pages 1–20, Gaitherburg, MD, 1992.

[10] Donna Harman, Wayne McCoy, Robert Toense, and Gerald Candela. Prototyping a Distributed Information Retrieval System Using Statistical Ranking. *Information Processing and Management*, 27(5):449–460, 1991.

[11] Shoshana Loeb and Douglas Terry. Special Issue on Information Filtering. *Communications of the ACM*, 35:26–81, 12.

[12] Kurt Maly, James French, Edward Fox, and Alan Selman. WATERS: The Wide Area Technical Report Service. In *Proc. 2nd Intl. World Wide Web Conf.*, Chicago, IL, October 1994.

[13] M. Mauldin and J. R. R. Leavitt. The Lycos Home Page: Hunting WWW Information, 1994. Available at http://lycos.cs.cmu.edu/.

[14] Zygmunt Mazur. On a Model of Distributed Information Retrieval Systems Based on Thesauri. *Information Processing and Management*, 20(4):499–505, 1984.

[15] Oliver A. McBryan. GENVL and WWWW: Tools for Taming the Web. In *Proc. 1st Intl. World Wide Web Conf.*, Geneva, Switzerland, May 1994.

[16] G. Salton. A Theory of Indexing. In *Regional Conference Series in Applied Mathematics No. 18, Society for Industrial and Applied Mathematics*, pages 1–56, Philadelphia, PA, 1975.

[17] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, NY, 1983.

[18] Anthony Tomasic and Hector Garcia-Molina. Query Processing and Inverted Indices in Shared-Nothing Text Document Information Retrieval Systems. *VLDB Journal*, 2(3):243–275, 1993.

[19] Anthony Tomasic, Hector Garcia-Molina, and Kurt Shoens. Incremental Updates of Inverted Lists for Text Document Retrieval. Technical Report STAN-CS-TN-93-1, Stanford University, 1993.

[20] Marc D. VanHeyningen. The Unified Computer Science Technical Report Index: Lessons in indexing diverse resources. In *Proc. 2nd Intl. World Wide Web Conf.*, Chicago, IL, October 1994.

[21] Charles L. Viles. Maintaining State in a Distributed Information Retrieval System. In *Proc. 32nd ACM Southeast Conf*, pages 157–161, Tuscaloosa, AL, March 1994.

[22] Justin Zobel, Alistair Moffat, and Ron Sacks-Davis. An Efficient Indexing Technique for Full-Text Database Systems. In *Proc. 18th Conf. Very Large Databases*, pages 352–362, Vancouver, BC, 1992.