are returned to the client are grouped based on the similarity of their metadata. There is currently no accepted way of ranking these groups. The client picks one of these groups, and downloads the associated file from a corresponding server [Rohrs, 2000, Rohrs, 2001].

There are other problems associated with the majority of the Gnutella (Version 0.4) based P2P information retrieval systems. These include global networking flooding and non-deterministic, poor accuracy search results. Since P2P information retrieval systems are, by definition, decentralized, for every search request, messages are sent to all nodes potentially possessing relevant documents. Since only limited search and document ranking capabilities exist at the nodes, any potentially relevant document is sent to the requesting nodes. Given the number of potentially relevant documents, the resulting network traffic overwhelms the network resources. This condition is referred to as global network flooding. To reduce the traffic, [Yu et al., 2003] developed a results filtering and merging technique where nodes collect information about their neighbors contents. Thus, when potential results to a request arrive at an intermediate node, only merged and filtered results are propagated back to the requesting node. A recent merging technique for hierarchical peer-to-peer networks is given in [Lu and Callan, 2004].

Another approach to reduce network traffic is to selectively transmit the request to only a limited set of target nodes. This approach suffers from potentially non-deterministic or poor result generation. Since only a limited number of nodes are accessed, depending on network conditions, different results may be obtained each time a request is made, hence the non-determinism. Furthermore, poor selection of the target nodes is likely to result in poor search accuracy. Hewlett Packard Company's PeerSearch [Tang et al., 2002, Tang et al., 2003] reduces the risk of poor target node selection by mapping the index and routing the queries according to a logical mapping of the vector space model retrieval coordinates onto the network. Thus, semantically near items should be mapped near each other, reducing the traffic.

Recently, a newer version of the Gnutella protocol [V0.6, 2004] was released. In this version, logical hierarchies are supported. That is, in the original P2P architecture models, all nodes were logically equivalent in function, and all information was decentralized. In this later protocol, some nodes act as *leaves* (client and server nodes) whereas other nodes serve as *directory nodes* providing no content but rather serving as a global directory for the content of their associated leave nodes. These hybrid networks, a composition of decentralized and centralized repositories, are providing additional opportunities for retrieval. Using these directory nodes, it is possible to more accurately access content and selectively choose proper target nodes. In [Liu and Callan, 2003], the authors propose and evaluate a P2P information retrieval system for such networks. The results presented demonstrated that by relying on the information in the directory nodes, network traffic is significantly reduced while accuracy is minimally compromised.

Some of the current academic P2P systems that are available are Edutella [Nejdl et al., 2002] and PeerDB [Ng et al., 2003]. These systems, however, are mainly focused on standardizing interfaces and query languages for Internet accessible services. Their goal is to create a P2P supercomputing or federated database system, not an information retrieval system.

Other research in P2P systems focuses on bandwidth efficiency. Rather than on relying on the Gnutella protocol, more intelligent routing protocols are presented [Ratnasamy et al., 2001, Sripanidkulchai et al., 2003, Stoica et al., 2001]. These efforts focus on improving response time, but do not improve the way that queries are answered and how results are ranked.

8.4.1 Example: Peer-to-Peer Information Retrieval System

In PIRS [Yee and Frieder, 2004], a P2P Information Retrieval System built on top of Gnutella, the primary goal is to enhance the search effectiveness by increasing the variety of system supported queries and improving the ranking mechanism. Improving the variety of queries that can be answered relies on building a corpus of metadata that reflects users' query patterns. Currently, the songs found on a typical Gnutella network contain metadata that are machine generated (e.g., using ID3 data [Nilsson, 2004] from Web sites such as freedb.org). A user who is unaware of the metadata annotation conventions, or deliberately poses a vague query, will not be able to find desired files. For example, a query for a "local Chicago band" will return nothing meaningful in the Gnutella network, whereas a query for " The Off-key Singing Trio" will.

PIRS solves this problem by forcing the client to randomly copy metadata from all the servers in the group from which the file was selected. This form of metadata copying significantly improves the number of queries that are satisfied.

PIRS ranks groups of files, not individual files, using the aggregate metadata of the group. It uses four ranking criteria:

- **Term Frequency** - measures the total number of times a query term appears in a group's metadata
- **Inverse Term Frequency** - measures term frequency, where each term's contribution is normalized by the inverse of its frequency over all groups
- **Precision** - measures the ratio of query term frequency and the total number of terms in the group
- **Group Size** - measures the total number of files in the group

Experimental results show that group size outperforms the other ranking criteria. There are two reasons for its effectiveness: a large group suggests that a file has a large degree of support for matching with a query; and a large group suggests that the file is popular in general, and is therefore a good download candidate. Overall, the use of random sampling, and group size ranking improves the query performance by an order of 30% over other combinations of metadata copying, and result ranking.

8.5 Other Architectures

In addition to the peer-to-peer architectures, some additional core distributed architectures exist. These include the shared disk architecture and the distributed data architecture. In each case, the notion is that the search is being done by multiple retrieval servers. If the search was being done by a single large machine with multiple processors then it would be a *parallel information retrieval search* (see Chapter 7).

8.5.1 Shared Disk Architecture

Even with numerous distributed processors, it is possible to link them all to a single shared disk. Incoming documents might be farmed out to the different servers for indexing, but all of the final index is stored in a single storage array. Queries can be processed by different servers, but they all access the index on the single storage array. The storage area becomes a significant point of contention, and it can result in degraded efficiency. The advantage is that even though the storage area can be a single point of failure, it is possible to use an additional storage array as a backup. The two storage areas can be synchronized at index time, and should a disk failure occur, the backup storage area can be immediately used.

If a processor crashes, it is only processing queries so users are not impacted at all. The entire index remains available to all users. Hence, reliability can be quite good with this architecture.

8.5.2 Distributed Disk Architecture

Here, each server has its own local disk, and the index is spread across the local disks. At index time, the document collection can be partitioned into the different servers, and they can all create the index in parallel and store their portion on their own local drives. Queries are also distributed across servers. Issued queries are sent to the processors that contain the posting list(s) that are requested. For common terms, this may be many different processors, but for an uncommon term, it may only be a small number of processors.

With this approach, there is no single point of resource contention; so it is extremely efficient. If a processor crashes, however, the portion of the index that is maintained by that processor is unavailable. For Web search applications this may be a reasonable tradeoff, and users do not need to be informed about the missing documents for a given query. For a mission-critical search engine, it is crucial to find all relevant documents. Simply ignoring some documents because a server is down is not acceptable. Building processor redundancy into this architecture requires doubling the number of machines, which may be very expensive.

Hence, the distributed architecture is typically viewed as more efficient than a shared disk architecture but less reliable.

8.6 Summary

This chapter focused on searching document collections that are physically distributed as well as search of document collections using a collection of separate machines. A Web search engine is inherently an example of a distributed information retrieval system since the actual documents are stored in servers around the world.

We started with a theoretical model of distributed information retrieval system and then moved into a brief discussion of recent work on distributed search strategies as well as very recent work on the use of peer-to-peer systems for information retrieval. Finally, we discussed other viable distributed architectures.

It is reasonable to expect that as cross-language information retrieval continues to grow and as more countries increase their stores of electronically available document collections, the need for highly effective, highly efficient distributed search systems will continue to increase.

8.7 Exercises

1 Develop a distributed IR algorithm that stores equally sized portions of an inverted index on separate machines. Compute the communications overhead required by your approach.

2 Describe the effects of document updates on a distributed information retrieval algorithm described in this chapter.

3 Recently Web search engines are facing the problem that developers of Web pages are adding terms that are commonly queried just to draw attention to their page. A user might add *Disneyland* to a page about *kitchen plumbing.* Develop a heuristic to circumvent this problem—talk about how your approach will avoid a reduction in effectiveness for a "normal" or untampered document collection.

Chapter 9

SUMMARY AND FUTURE DIRECTIONS

We described a variety of search and retrieval approaches, most of which primarily focused on improving the accuracy of information retrieval engines. Unlike other search and retrieval domains, e.g., traditional relational databases, the accuracy of retrieval is not constant. That is, in the traditional relational database domain all techniques result in perfect accuracy. Hence, the main concern, in terms of performance evaluation, is the overall system throughput and the individual query performance.

In the information retrieval domain, accuracy varies as the associated precision and recall measures of all engines are both approach and data dependent. Thus, all information retrieval performance evaluation must account for both the resulting accuracy, as well as the associated processing times. In both database and information retrieval systems performance evaluation, commonly referred to as *benchmarking*, must also take storage overhead into account. Given the continuing improvements in storage technology coupled with the ongoing reduction in cost, relatively little attention is focused on storage overhead reduction as compared to improving computational time, and where appropriate, accuracy demands.

To assess the performance of database systems, many benchmarks were developed. Many of these benchmarks are in commercial use. Examples include the TPC family of benchmarks [Kohler, 1993]. Until the early 1990's, little emphasis was placed on the development of benchmarks for uniform evaluation of the performance of information retrieval approaches or engines. The datasets used in the evaluation of information retrieval systems were small in size, often on the order of megabytes, and the mix of queries studied were limited in domain focus, number, and complexity.

In 1985, Blair and Maron [Blair and Maron, 1985] authored a seminal paper that demonstrated what was suspected earlier: performance measurements obtained using small datasets were not indicative for larger document collections. In the early 1990's, the United States National Institute of Standards and Technology (NIST), using the text collection created by the United States Defense Advanced Research Project Agency (DARPA), initiated a conference to support the collaboration and technology transfer between academia, industry, and government in the area of text retrieval. The conference, named the Text REtrieval Conference (TREC) aimed to improve evaluation methods and measures in the information retrieval domain by increasing the research in information retrieval using relatively large test collections on a variety of datasets.

TREC is an annual event held in November at NIST. Over the years, the number of participants has steadily increased and the types of tracks have greatly varied. In its most recent 2003 incarnation, the twelfth conference, TREC consisted of six tracks, namely Genomics, HARD, Novelty, Question Answering, Robust Retrieval, and Web. The specifics of each track are not relevant since the tracks are continuously modified. Suffice to say that the type of data, queries, evaluation metrics, and interaction paradigms (with or without a user in the loop) vary greatly. The common theme of all tracks is to establish an evaluation corpus to be used in evaluating search systems.

Conference participation procedures are as follows. Initially a call for participation is announced. Those who participate eventually define the specifics of each task. Documents and topics (queries) are procured, and each participating team conducts a set of experiments. The results are submitted for judgment. Relevance assessments are obtained, and the submitted results are evaluated. The findings are evaluated, summarized, and presented to the participants at the annual meeting. After the meeting, all participants submit their summary papers, and a TREC conference proceeding is published.

Early TREC forums used data on the order of multiple gigabytes. Representative collection statistics are listed in Table 9.1. A sample document from the above collection is presented in Figure 9.1. A sample query is illustrated in Figure 9.1.

Today, the types of data vary greatly, depending on the focus of the particular track. Likewise, the volumes of data vary. At the writing of this second edition, a terabyte data collection is proposed for one of the 2005 TREC tracks with a preliminary collection somewhat in excess of 400 GB to be used in 2004. Thus, within roughly a decade, the collection sizes will grow by three orders of magnitude from a couple of gigabytes to a terabyte. This growth of data might necessitate new evaluation metrics and approaches.

Throughout its existence, interest in TREC activities has steadfastly increased. With the expanding awareness and popularity of distributed information retrieval engines, e.g., the various World Wide Web search engines, the number of academic and commercial TREC participants continues to grow. Given this increased participation, more and more techniques are being developed and evaluated. The transfer of general ideas and crude experiments from TREC participants to commercial practice each demonstrates the success of TREC.

Table 9.1. Size of TREC data

Disk	Collection	Size (MB)	Number of Documents	Median $\frac{Terms}{Doc}$	Mean $\frac{Terms}{Doc}$
1	*Wall Street Journal*, 1987—1989	267	98,732	245	434.0
1	*Associated Press*, 1989	254	84,678	446	473.9
1	*Computer Select*, Ziff-Davis	242	75,180	200	473.0
1	*Federal Register*, 1989	260	25,960	391	1,315.9
1	abstracts of US DOE	184	226,087	111	120.4
2	*Wall Street Journal*, 1990—1992	242	74,520	301	508.4
2	*Associated Press*, 1988	237	79,919	438	468.7
2	*Computer Select*, Ziff-Davis	175	56,920	182	451.9
2	*Federal Register*, 1988	209	19,860	396	1,378.1
3	*San Jose Mercury News*, 1991	287	90,257	379	453.0
3	*Associated Press*, 1990	237	78,321	451	478.4
3	*Computer Select*, Ziff-Davis	345	161,021	122	295.4
3	*US Patents*, 1993	243	6,711	4,425	5,391
4	*Financial Times*, 1991—1994	564	210,158	316	412.7
4	*Federal Register*, 1994	395	55,630	588	644.7
4	*Congressional Record*, 1993	235	27,922	288	1,373

Over the years, the raw average precision numbers presented in the various TREC proceedings initially increased and then decreased. This appears to indicate that the participating systems have actually declined in their accuracy over some of the past years. In actuality, the queries and tasks have increased in difficulty. When the newer, revised systems currently participating in TREC are run using the queries and data from prior years, they tend to exhibit a higher degree of accuracy as compared to their predecessors. Any perceived degradation is probably due to the relative complexity increase of the queries and the tasks themselves.

We do not review the performance of the individual engines participating in the yearly event since the focus of this book is on algorithms, and the effects of the individual utilities and strategies are not always documented. Detailed

```
<DOC>
<DOCNO>WSJ880406-0090 </DOCNO>
<HL> AT&T Unveils Services to Upgrade Phone Networks Under Global Plan </HL>
<AUTHOR> Janet Guyon (WSJ Staff) </AUTHOR>
<DATELINE> NEW YORK </DATELINE>
<TEXT>
American Telephone & Telegraph Co. introduced the first of a new generation of phone
services with broad implications for computer and communications equipment markets. AT&T
said it is the first national long-distance carrier to announce prices for specific services under
world-wide standardization plan to upgrade phone networks. By announcing commercial
services under the plan, which the industry calls the Integrated Services Digital Network,
AT&T will influence evolving communications standards to its advantage, consultants said,
just as International Business Machines Corp. has created de facto computer standards favoring
its products. ...
</TEXT>
</DOC>
```

Figure 9.1. Sample TREC document

```
<top>
<num> Number: 168
<title> Topic: Financing AMTRAK
<desc> Description:
A document will address the role of the Federal Government in financing the operation of the
National Railroad Transportation Corporation (AMTRAK)
<narr> Narrative:
A relevant document must provide information on the government's responsibility to make
AMTRAK an economically viable entity. It could also discuss the privatization of AMTRAK
as an alternative to continuing government subsidies given to air and bus transportation with
those provided to AMTRAK would also be relevant.
</top>
```

Figure 9.2. Sample TREC query

information on each TREC conference is available in written proceedings or on-line at: trec.nist.gov.

TREC, although successful, does have its shortcomings. As noted, performance evaluation in retrieval systems involves both accuracy and performance assistance. TREC, however, only evaluates accuracy, paying little if any, significance to processing times and storage overheads. In terms of relevancy (accuracy), common TREC criticism focuses on the means of judging document-to-query relevancy.

Given the limited number of human document judgment analysts available to NIST, pooling is used to determine the relevant documents. Pooling, as now used in TREC [Harman, 1995], is the process of selecting top-ranked documents obtained from multiple engines, merging and sorting them, and retaining the remaining unique document identifiers as relevant documents (i.e., removing the duplicate document identifiers). Although relatively effective, pooling does result in several false-negative document ratings. To avoid such problems, recent work describes methods which avoid the need for pooling [Sanderson and Joho, 2004].

We also note that a new effectiveness measure has been proposed to be more resilient to problems with incompleteness in a pool. Pooling assumes that *most* relevant documents will be found. When that is not the case, average precision is not robust [Buckley and Voorhees, 2004]. Furthermore, average precision does not include a user preference. A document not found is treated the same as a document that is not relevant. A preference based measure focuses on the number of times judged nonrelevant documents are retrieved before relevant documents. Simply counting the number of non relevant before we hit a relevant document is not sufficient because of the vast differences in relevant documents available to a query. A query with one hundred relevant documents has a much higher chance of hitting a relevant document than one that has only one or two relevant documents. To normalize for the number of relevant documents, the following measure is proposed:

$$bpref = \frac{1}{r}\sum_{r}\left(1 - \frac{|n \ ranked \ higher \ than \ r|}{R}\right)$$

Another recently described form of evaluation uses term relevance as opposed to document relevance. Here, a list of relevant terms is identified for a given query and effectiveness is measured based on a system's ability to find terms in the list. Since, a set of terms can be defined for a query without detailed knowledge of a test collection, this approach has the potential of scaling to very large document collections [Amitay et al., 2004].

We note that the Cross-Language Evaluation Forum (CLEF) has followed the basic TREC style but focuses on cross-lingual evaluation (see Section 4.1). The CLEF Web site is **clef.iei.pi.cnr.it:2002**. Additionally, a new test collection of conversational speech was recently developed [Oard et al., 2004].

In spite of all the past successful research efforts, the domain of information retrieval is still in its infancy. Twenty years ago, the number of retrieval strategies could be counted on one hand. Most of the research literature focused on the four key retrieval strategies: the vector space, probabilistic, Boolean, and fuzzy-set. Since we developed our first edition of this book, language models were applied to the problem and the result is a ninth retrieval strategy.

Until recently, distributed information retrieval was only of theoretical interest. With the expansion of personal Internet use and the advent of the World Wide Web (WWW), distributed information retrieval, specifically search and retrieval of information across the WWW, is a daily practice.

Cross-language retrieval was just getting underway when we went to press with the first edition. Now it is a fairly mature area. It has taken us one step closer to a search environment that would allow a user to query a world wide document collection and obtain results in a single language.

We should also note that search of HTML pages is only one restricted type of search. Most large companies have search problems that cover large bodies of texts without hyperlinks (e.g.; Word documents, PowerPoint presentations, PDF files). The techniques that work well for Web search do not necessarily perform well for these data.

In terms of the research community, heightened interest is best demonstrated by the increased popularity of the NIST TREC activities. In its initial years, the number of participants in the TREC activities numbered less than thirty for most tasks. In the sixth NIST TREC meeting, the number of participants exceeded fifty. In the twelfth offering, the number of participants exceeded 100. The conference started with tests of only two Gigabytes of text (an amount which was very difficult for many researchers) and is now gearing up for a Terabyte of text.

Given the growing interest, future advances are clearly on the horizon. The question is which areas still need further investigation. We project future research using the same paradigm used throughout the book. That is, first we address strategies, followed by utilities and efficiency concerns. Issues involving parallelism and distributed processing conclude our projections.

Additional data strategies are still required. In the TREC activities, the average precision numbers rarely reach the forty percent mark for any task. Significantly improving these numbers requires new insight and potentially a new strategy. The past several years have resulted in a steady improvement in retrieval accuracy, but current results are still unacceptable. It is unlikely that even this continued improvement will result in significant strides to sufficiently improve retrieval accuracy. This is especially true when faced with vastly larger data sets. It is reasonable to suspect that simple pattern matching approaches will continue to stay at the existing plateau observed in TREC during the last two or three years. To go beyond this will likely require incorporation of more complex natural language processing. At present, recent work on information extraction and "light parsing" are just now becoming computationally feasible.

Additional strategies are also required to cope with the diversity of data presently available on-line. Throughout this book, we addressed only text oriented data. Given the adage that a picture is worth a thousand words, one must find a way of extracting and integrating the thousands of words portrayed by an image. Currently, information retrieval models do not support this. There are efforts that address the image integration issue, for example, IRIS (adapted as the IBM ImageMiner Project) [Alshuth et al., 1998]. However, they still do not fully integrate structured, text, and image data into a cohesive environment. It is reasonable to expect that the future will require an extended corpus consisting of integrated text with images. Such a corpus will make it possible to evaluate progress of new text and image retrieval algorithms.

It is possible to represent information retrieval processing utilities on a continuum where the two extremes are simple pattern matching and full natural language text processing. Currently, the majority of the utilities fall closer to the simple pattern matching end of the continuum. For example, both passage-based and n-gram techniques clearly focus on purely pattern matching analysis. Semantic networks and parsing techniques more closely align with natural language processing, but clearly do not support full content analysis as expected from natural language processing. It is our belief that to significantly increase the accuracy of retrieval, the semantic meaning of the text, in contrast to its denotational, or even worse, purely its character representation, must be extracted. When we went to press with the first edition, text extraction was a relatively new field. Now, there are a variety of commercial text extractors. The question still remains: How can these tools be used to improve information retrieval?

Parallel processing architectures are now widely available and are in daily use. They are no longer just research engines. Even our personal computers are configured as parallel processing engines. Thus, information retrieval applications must be developed to harness this parallel processing capability. In Chapter 7, we described some of the ongoing parallel processing efforts. None of these efforts, however, have demonstrated scalability to the thousands of nodes. None can handle a diversity of data formats, support multi-language retrieval, efficiently support all of the described retrieval strategies and utilities, provide multi-user concurrency with on-line recovery, and support a "plug-and-play" composition of strategies and utilities environment. Furthermore, with the diversity of the underlying models of parallel architectures, even if some solutions to the above concerns are available, they do not seamlessly port across multiple parallel architectures. Clearly, in the realm of parallelism in information retrieval, there is a wide area for further investigation.

With the continued advances in wireless technology, data are available not only on host computers, but also on mobile computing devices worldwide. This distributed nature introduces several issues not previously of much concern to the information retrieval domain. For example, due to the portable nature of the storage devices, most of the data are available only at uncertain time intervals. Furthermore, each search site has access to only limited information and this information can change rapidly. Thus, distributed information retrieval algorithms must account for these constraints. Some ongoing research efforts in the domains of distributed operating and database systems focus on related issues. An adaptation of some of the results from such efforts might be appropriate. To date, no information retrieval research efforts address these concerns.

Throughout this book, we have advocated a plug and play architecture for information retrieval. We described strategies, utilities, efficiency considerations, integration paradigms, and processing topologies for information retrieval. The primary problem for future information retrieval research investigation is: How does one achieve synergy in the composition of all of these factors?