Information Retrieval: Research into New Capabilities

by Edward A. Fox

Microcomputers and optical discs have captured the imagination of computer scientists and users alike. What is needed is a coalescing of these two developments with the fruits of retrieval research, to make a quantum leap beyond current systems in all three areas.

In 1945, Vannevar Bush published his seminal article "As We May Think," in which he envisioned a device called a "memex" that would transform a desk into a portal to the world's stored knowledge. It would be possible not only to access information but to record one's trail of investigations and share those associations with others.

In the past four decades, researchers in the little-known field of information retrieval have struggled with technological and scientific issues in their attempt to create systems with some of the capabilities of the memex. While Bush's dream was initially described in terms of microfilm and mechanical devices, today's designers have microcomputers, fiberoptic networks, and laser discs. Technologically, we are much closer to the paperless society described by Lancaster (1978). More important in many ways, however, is the fact that computer science has emerged as a key area of research in the new information age and that the field of information retrieval has matured into a cross-disciplinary investigation of the problems of text analysis, indexing, knowledge representation, storage, access, and presentation.

For years, work in information retrieval has been limited by the available tools. Today, computer hardware with tremendous speed and storage capacity has become easily affordable. Now there is a chance to use innovative approaches to develop new retrieval software and to test that out with large collections of information.

Microcomputers and optical discs have captured the imagination of computer scientists and users alike. What is needed is a coalescing of those two developments with the fruits of retrieval research, to make a quantum leap beyond current systems in all three areas. The result will be an exciting range of products bringing us another step closer to what Vannevar Bush envisioned.

Focus

Papyrus, the world's first paper, revolutionized the recording of information. Today, CD ROM is starting a similar revolution. Providing access to that information, however, becomes even more important, due to the sheer volume of material involved.

This chapter focuses primarily on research issues relating to access methods and techniques developed in the area of information retrieval. The discussion will center on models, on experiments testing system effectiveness, and on indexing/retrieval/interface methods. For additional information on these subjects, you can consult works by authors such as Heaps, Van Rijsbergen, Salton, and McGill, listed in the bibliography at the end of the chapter.

Research Issues

Information retrieval (IR) is a cross-disciplinary field investigated by librarians, linguists, psychologists, and computer or information scientists. Two of the closest areas to IR are the study of database management systems (DBMS) and artificial intelligence (AI). What is distinctive about IR is the need to work with:

- Unstructured data, unlike the structured data of database systems.
- Text, possibly in concert with numeric and image data.
- Very large collections, often larger than those handled by a DBMS, and much more heterogeneous than those investigated by AI researchers.
- Effectiveness of access, measured experimentally.

Certainly, DBMS and AI researchers confront some of these issues, but they are not necessarily of central interest to them. The second point is clearly important for linguists; the third, for library and information scientists; and the fourth, for cognitive psychologists.

Representation

How information is viewed by system designers and users is one of the key differences between the areas of DBMS, IR, and AI. Figure 1 illustrates the distinction in terms of a collection of bibliographic citations.

BIBLIOGRAPHIC CITATIONS			
TITLE	AUTHOR	PUBL.	YEAR
Title1	Author1	New P.	1984
Title 1	Author2	New P.	1984
Title 2	Author3	Publ. X.	1985

```
.T
Title of the book or article
.A
First Author, Jr.
Second Author, II
.P
New Publishing Co.
.Y
1984
```

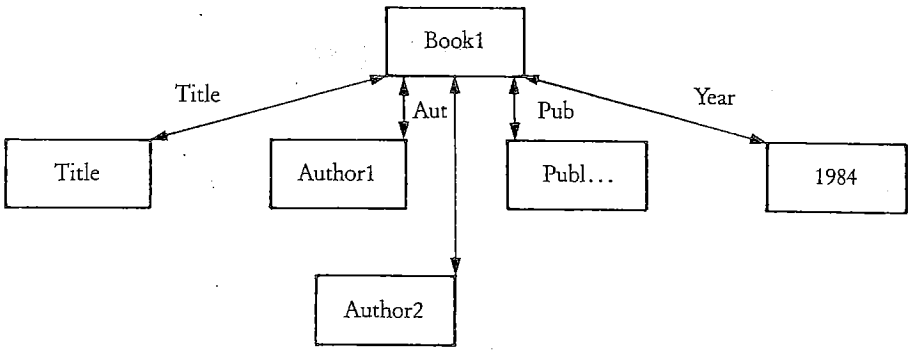

Figure 1. Representations of citation data in database, IR, and AI systems

A good deal of attention has been given to integrating IR and DBMS technology. Dattola (1979) developed the FIRST system, which merged a network database and a clustered retrieval system. Crawford (1981) advocated the adaptation of the relational model for IR purposes, and with Macleod (1983) viewed IR as a database application. Developers of database systems, such as Stonebraker (1983), have made special adaptations to better support document processing. Although some DBMS techniques, such as B-trees, have been integrated into IR systems, an elegant merger of the concepts has not yet occurred.

Other work has focused on the nontextual data relating to documents and shown the value of that additional type of information. Citation data can lead to cocitation diagrams, such as those developed by Small (1980), which highlight the interrelationships between areas in a field. Bibliographic coupling, explored by Kessler, is also of value. When these relationships are used in concert with descriptive and textual data, the resulting representation, according to results with two special test collections which this author developed, can lead to even more effective retrieval.

Text Processing

Text appears in manuscripts, journals, reference works, periodicals, reports, newspapers, correspondence, and other forms. Computer methods that aid in the creation and layout of text, such as word processing, electronic messaging, and photocomposition, have led to the increased availability of machine-readable texts. Standards work, as in the Electronic Manuscript Project, will accentuate this trend. Attendant issues of analyzing, indexing, storing, and accessing text have been central to much of the work in modern IR.

Computer methods that aid in the creation and layout of text, such as word processing, electronic messaging, and photocomposition, have led to the increased availability of machine-readable texts.

The simplest way for a computer to access text is to search it sequentially, much as people scan the words on a page. When people search through text collections they are able to decide whether located items are relevant to their particular search need. Computerized retrieval systems should find exactly those items that are relevant—ignoring items that are nonrelevant. That is to say, they should have high precision as well as high recall. Clever algorithms, like that of Boyer and Moore (1977), can speed up the search process, and further improvements are possible with special hardware. But the simple sequential search becomes ineffective as the query increases in complexity and the text collection increases in size. However, creating complex models of text documents by analyzing and indexing large collections can allow them to be efficiently searched.

Models

Much of the research in IR has been based on four different models of the field: browsing, Boolean and p-norm, vector and probabilistic, and artificial intelligence. Traditionally, the Boolean model has been most widely upheld. The vector and probabilistic models have shown promise in laboratory tests. With recent advances in display and pointing devices, browsing models have been advanced. Finally, due to the promise and popularity of AI, a good deal of attention has been given to AI-based models. Each of these models is discussed below, in order of complexity.

Browsing Models

In libraries, people work with the card catalog (or its equivalent) and/or browse in the stacks. When using reference works, people frequently follow implicit or explicit pointers or skip around looking for particular items. Now that large high-resolution displays are available for computers, effective use of such two-dimensional devices, as in the Caliban system, is of particular interest.

The Smalltalk-80 system, developed by Goldberg et al. (1982), has been implemented to provide such an environment. Since hierarchical organizations of knowledge are commonplace, the language and displays support that perspective. An example of the system browser is shown in Figure 2. Once Smalltalk is thoroughly understood, program development for broad classes of applications becomes especially convenient.

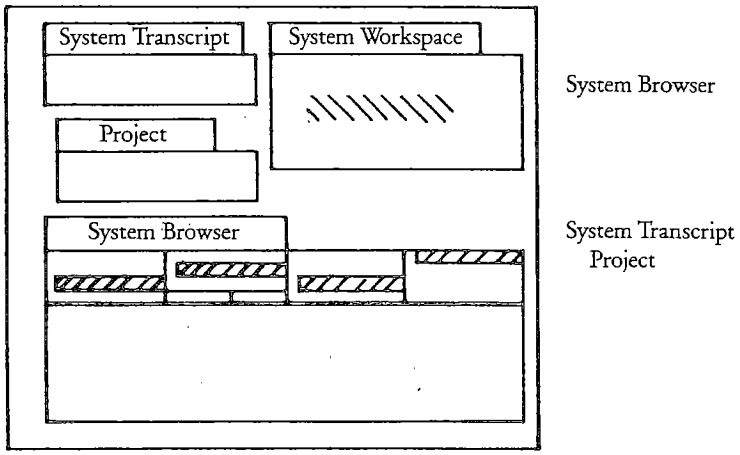

Figure 2. Smalltalk-80 with browser

Weyer explored the value of a browsing model for searching books, as portrayed in Figure 3. The key point is that users can see several adjacent displays supporting various types of access to a book and can easily explore useful links. Responses to a question can be located using the table of contents, the index, or other pointers, or by the aid of text searching.

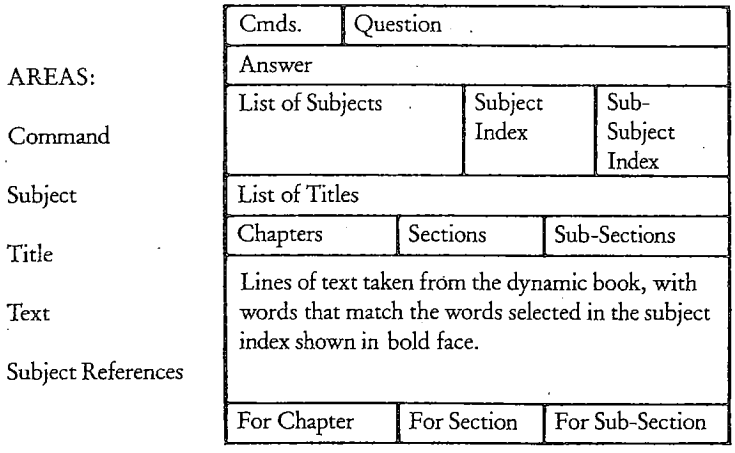

Figure 3. Dynamic book

On a somewhat larger scale, electronic encyclopedias, such as the Grolier encyclopedia described by Cook (1984), provide browsing-based access to large reference works. As can be seen in Figure 4, all the cross references found in an encyclopedia can supplement what is available in a dynamic book. Furthermore, special capabilities can be built into such systems to understand units of measure, foreign words, or alternative map presentations. Lessons, simulations, and other enhancements can also be incorporated.

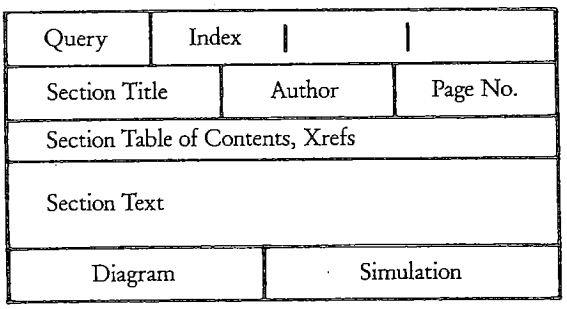

Figure 4. Electronic encyclopedia

For large collections of documents, integrating browsing with other access methods is of particular appeal. Caliban is the first retrieval system with that orientation. It employs some of the most useful search techniques, along with multiwindow display and browsing. The appeal of this model has encouraged other researchers, such as Thompson (1985) and Croft (1978, 1982), to incorporate some of the ideas of Caliban in more recent systems that employ AI methods as well.

Boolean and P-Norm Models

Most retrieval systems require users to describe their search interest in terms of a Boolean expression, where key words or other atomic elements are combined into clauses, and clauses are combined with other components, by AND, OR, or NOT operators. Figure 5 illustrates this approach. Part *A* is an English language statement indicating an area that might be of interest to a reader of this book. Part *B* is a typical expression of the same statement as a Boolean query. For those with a computer science bent, part *C* uses prefix notation to give an equivalent form that can be directly mapped into the tree structure in part *D*, which illustrates the relationship of the various concepts.

It should be noted that the original form of part *A* could not be directly utilized by a computer. Indeed, it is difficult to construct good Boolean

A)

INFORMATION RETRIEVAL SYSTEMS WITH OPTICAL DISC STORAGE SUCH AS CD ROM ATTACHED TO STANDARD PERSONAL COMPUTER

B)

(IR OR (information AND retrieval) OR (document AND retrieval)
AND
(CD ROM OR CDROM OR (CD AND ROM) OR (compact AND disc AND (ROM OR read-only OR memory or stor*)))
AND
(PC OR PC-AT OR PC-XT OR (personal AND computer AND IBM))

C)

AND (OR (IR, AND (information, retrieval), AND (document, retrieval)), OR (CD ROM, CDROM, AND (compact, disc, OR (ROM, read-only, memory, stor*))), OR (PC, PC-AT, PC-XT, AND (personal, computer, IBM)))

D)

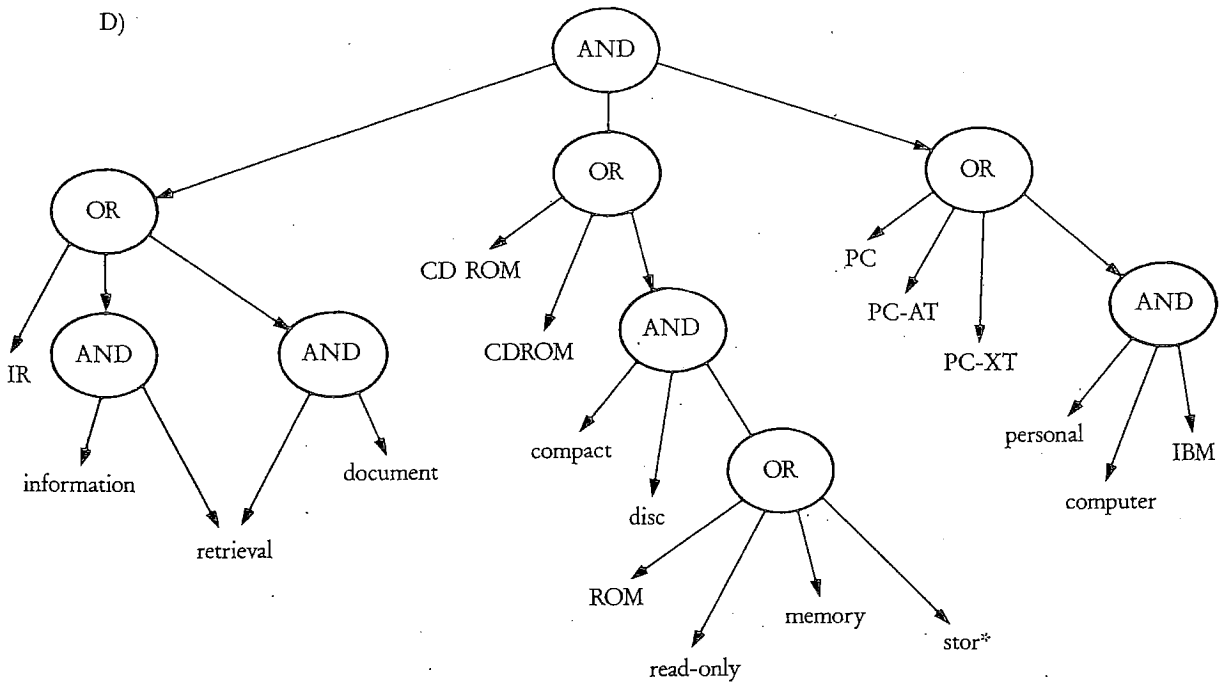

Figure 5. Transformations of Boolean query representations

queries, often requiring the assistance of trained search intermediaries who add terms not originally included (e.g., document), provide synonyms (e.g., PC) or alternate spellings (e.g., CDROM), truncate words to allow for morphological variants (e.g., stor*), drop function or high-frequency terms (e.g., with, systems), and insert Boolean operators. Of the available Boolean operators, AND and OR are the most commonly used. OR is required when any type of "searchonym" (as coined by Attar and Fraenkel [1977], a term synonymous for searching purposes) is involved, while AND must be used sparingly so as not to diminish recall. It is rarely necessary to use NOT. When high precision is needed, metrical, proximity, or adjacency operators are also employed, since it is less likely for terms to appear near each other than to simply occur in the same document. The entire process of carrying out a Boolean search is summarized in Figure 6.

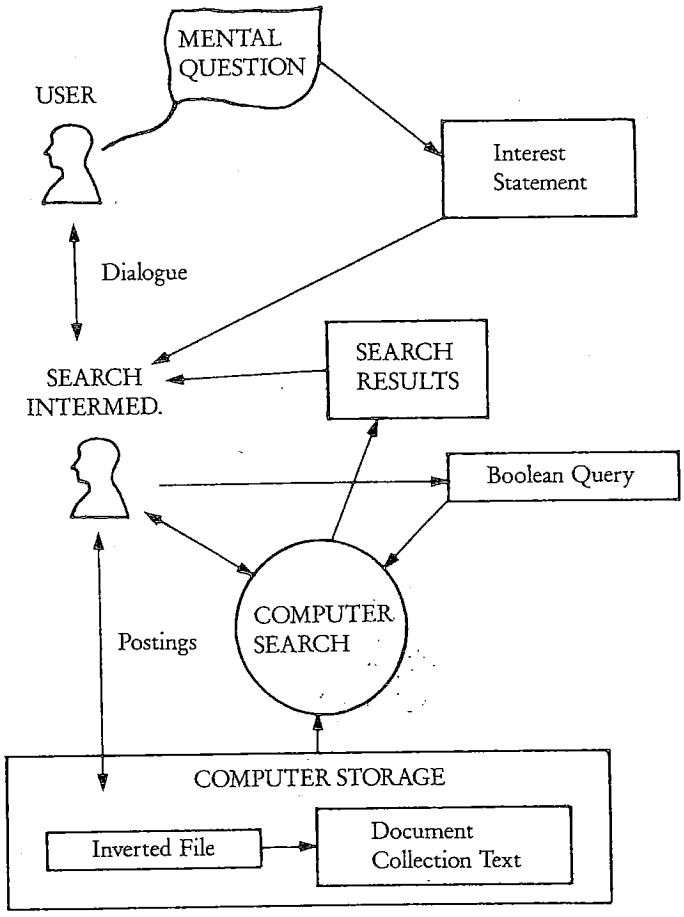

Figure 6. Boolean query formation and inverted file search

In actuality, there is usually a good deal of interaction between the end user and the searcher, and between the searcher and the system, in order to construct a good query. Part of that latter dialog is shown in Figure 7.

> information		> CD ROM	
1) INFORMATION	2300	9) CD ROM	421
> retrieval		> CDROM	
2) RETRIEVAL	749	10) CDROM	121
> document		> compact	
3) DOCUMENT	892	11) COMPACT	82
> 1 AND 2		> disc	
4) 1 AND 2	539	12) DISC	1583
> 3 AND 2		> 11 AND 12	
5) 2 AND 3	372	13) 11 AND 12	48
> IR		> ROM	
6) IR	452	14) ROM	567
> 4 OR 5		> read-only	
7) 4 OR 5	627	15) READ-ONLY	391
> 7 OR 6		> memory	
8) 6 OR 7	704	16) MEMORY	3865

Figure 7. Initial section of dialog for Boolean query processing

A retrieval system accepting Boolean queries typically includes an inverted file (also called an inverted index) which supplies the postings information shown in Figure 7. Thus, whereas the document file is stored as a set of documents, each of which has words, the inverted file (IF), depicted in Figure 8, turns that organization upside down.

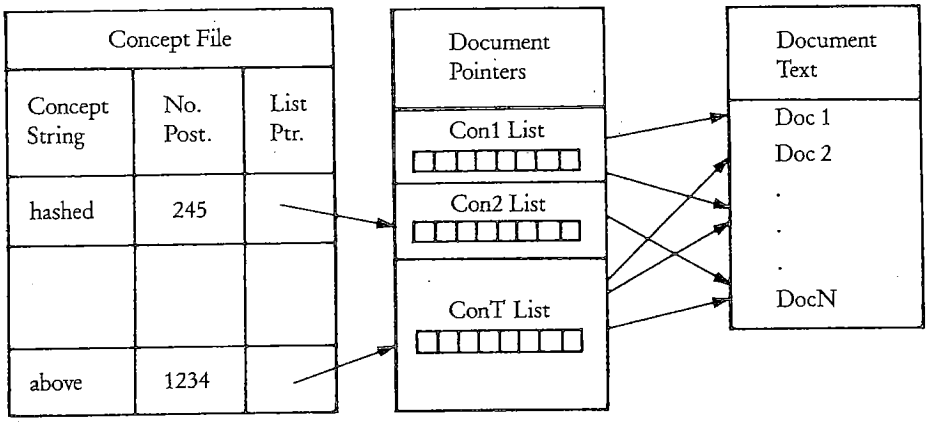

Figure 8. Simple inverted file organization

For each term (e.g., keyword, special phrase, thesaurus class), there is a list of all documents in which it appears, along with the length of that list (i.e., the number of postings). From Figure 7 it can be seen that a searcher will not only consider the words given and their semantic relationships but will also note how frequently words occur, or how commonly they occur with other words. A Boolean query thus will often evolve in steps; and although the final query will probably retrieve a reasonable number of hits, it may have a very complex syntactic and bizarre semantic structure.

The simplest way for a computer to access text is to search it sequentially, much as people scan the words on a page.

In spite of the ubiquitous nature of Boolean query systems for IR, they have many limitations and problems. As can be seen in Figure 7, it is difficult to identify the correct-size set of documents that should be retrieved. When all terms thought to be useful are put together into a suitable query, the resulting retrieved set may be much too large or too small. Furthermore, if there is a large set, the documents are usually randomly ordered, so the most useful one may well be at the end.

The SIRE system was devised by McGill, Noreault, and Koll in part to explore the effects of ranking the set of documents retrieved by a Boolean query. In addition to storing the list of documents in which a term occurred, the number of times a term appeared in each document was recorded in the inverted file. A variety of formulas for ranking the retrieved documents were compared by McGill et al. (1976) in order to group similar schemes and identify the best ones.

Even in SIRE, however, the searcher cannot specify which terms are most important. One notation for expressing relative importance was proposed by Bookstein (1980). When Boolean logic is generalized so that there are degrees of truth in the range of 0 to 1, then a document can be indexed by a term to a partial degree, and truth values can be viewed as measuring the similarity between a query and each document. Figure 9 illustrates the value of having such *fuzzy* sets rather than simple Boolean sets of retrieved documents. It is likely that relevant documents will have higher similarity and so will appear closer to the top of the list; more documents will typically be retrieved as well, aiding with recall.

A. FUZZY SET

Document	Similarity
ID001	1.0
ID742	.834
ID819	.632
•	•
•	•
•	•
ID253	.104
ID006	0.0
•	•
•	•
•	•
ID997	0.0

B. BOOLEAN

Document	Similarity
ID001	1.0
•	•
•	•
•	•
ID029	1.0
ID003	0.0
•	•
•	•
•	•
ID997	0.0

Figure 9. Sets of retrieved documents

Such a ranking can be produced when Boolean queries are interpreted according to the p-norm scheme explored by Salton, Fox, and Wu (1983). In addition to having relative weights on query terms, and degrees of indexing (in range 0 to 1) on terms in each document, one can vary the strictness of interpretation of the OR and AND operators (in similar fashion to the softening suggested by Paice [1984]). That is to say, the conjunction of a number of terms will retrieve documents where all terms need not be present, albeit with a similarity less than 1. Also, the disjunction of terms will lead to higher similarity when several of the terms are present as opposed to when only one is present. A p-value can be chosen in the range of 1 to infinity, where 1 gives the least strict interpretation. The p-value identifies which in the Lp family of norms is employed to measure distance from the ideal points (since OR queries should be far from the 0 point, and AND queries should be close to the 1 point), as can be seen in Figure 10. Note that p = 2 specifies standard Euclidean distance. These curves, developed by this author, show connect points for documents that have equal similarity to the given two-term queries. They demonstrate the range in strictness from the p = infinity case, where AND is viewed as MIN and OR is viewed as MAX, to the p = 1 case where AND = OR = AVERAGE.

Experiments by this author indicate that the p-norm scheme leads to more effective retrieval than traditional Boolean systems. P-norm queries can also be automatically constructed from simple lists of keywords. In addition, when a user is presented with the first 10 or 20 documents retrieved by a p-norm or Boolean search and indicates which of those are relevant,

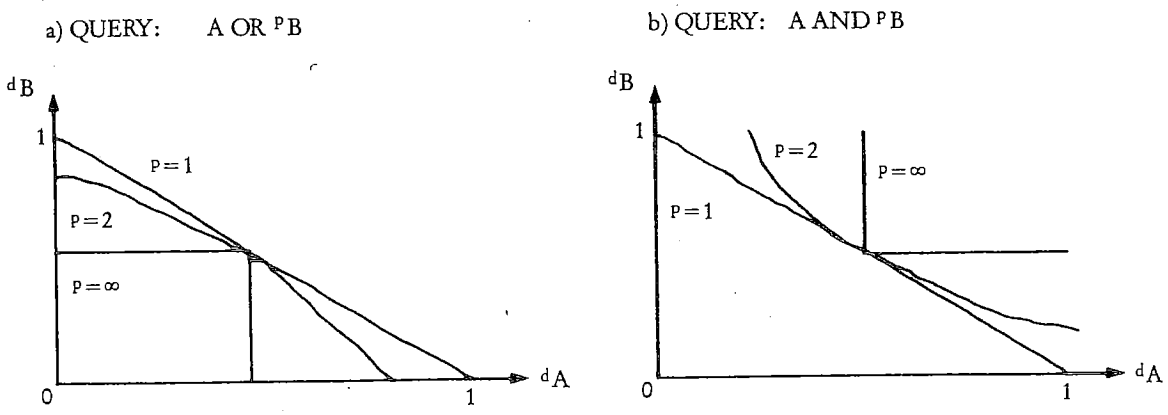

Figure 10. P-norm equisimilarity contours

automatic construction of a new feedback query is possible. Experiments by this author and by others at Cornell have shown that such a query is often better than the original query.

Vector and Probabilistic Models

The vector and probabilistic models developed out of early investigations into the statistical characteristics of text collections. Specifically, the vector space model proposed by Salton, Wong and Yang (1975) is based on the observation that the frequencies of occurrence of terms are indicative of their importance. Similarly, the probabilistic model, proposed at roughly the same time by Robertson and Sparck Jones (1976) and by Salton, Yang and Yu (1975), relies on Bayesian theory. The presumption is that term importance can be estimated as a result of contrasting the occurrence characteristics of text terms in a feedback set of relevant documents with occurrence values in the rest of the collection. Both of these models are discussed at length by Van Rijsbergen (1979); the description below deals with key features.

Figures 11 and 12 illustrate the vector space model, giving examples of spaces for three-dimensional terms and for T-dimensional terms, respectively.

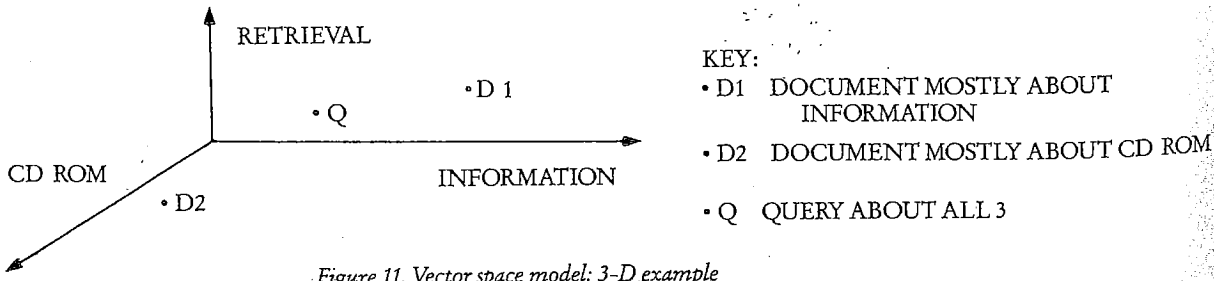

Figure 11. Vector space model: 3-D example

Three of the terms from the query in Figure 5A are chosen in Figure 11, where documents are represented by points determined by the number of times each of the terms occurs. The query Q can likewise be located in the space. Relative weights on the terms allow one to indicate which are most important. Retrieval can be viewed as locating documents that are near the query, based on a suitable definition of similarity.

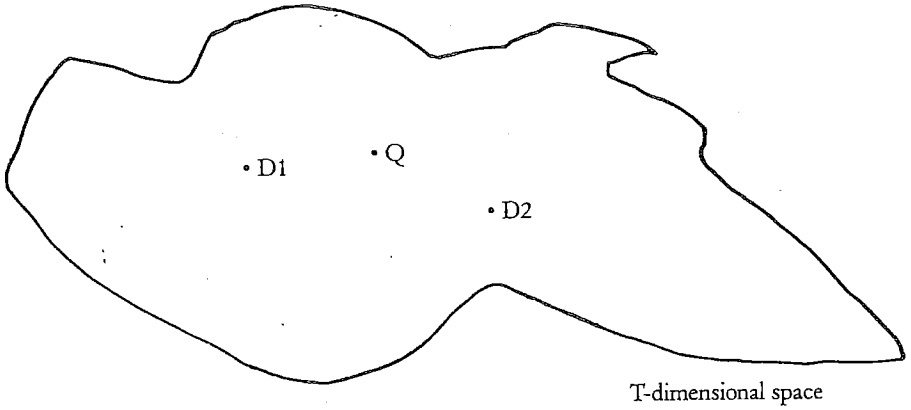

Figure 12. Vector space model: T-D example

In Figure 12, an idealized view of the T-dimensional space determined by the T different concepts in the collection, documents and queries can both be placed. Similarity can then be computed, for example, as the cosine of the angle between a query and document vector.

Figure 13 shows one way to represent the vector space, using matrix notation. Each of the T concepts present is associated with a column, and each of the N rows defines a collection document. The value for document I and term J, d_{ij}, indicates the importance of that term's appearance.

If the cosine similarity measure is employed, TF × IDF weights are easy to compute and give good performance; they are the product of the number of times the term appears in the document (the TF, or term frequency, component) and the inverse document frequency (IDF, figured as the log of N/DFj, where DF is the number of documents in which the term appears). A similar value has been used by this author and others for p-norm computations, except that normalization to the range of 0 to 1 is required.

TERMS

	1		J		K			T
1	d-11							d-1T
I	d-i1		d-ij		d-ik			d-iT
N	d-N1		d-Nj		d-Nk			d-NT

DOCS

Figure 13. Document-term collection matrix

In any case, the document-term matrix in practice is very large, but since it is sparse it can be compactly represented. If one stores data by column, an extended inverted file like that found in SIRE results. Storing by row results in a file of document vectors.

Figure 14 idealizes the vector feedback process first developed by Rocchio (1971). A query, Q0, is transformed into a vector, and then a set of nearby documents is initially retrieved. The searcher identifies the (3, in this case) relevant documents. A new feedback query, Q1, is constructed from Q0 by moving closer to the relevant retrieved documents. Presumably, Q1 will do a better job than Q0 in retrieving relevant documents.

Metrical feedback, advocated by Attar and Fraenkel (1977), follows a similar model but also considers the number of intervening words between occurrences of terms that are to be added to the original query. Probabilistic feedback has a similar effect, but the construction of Q1 is done based on a term relevance (Robertson and Sparck Jones, 1976) or term precision formula (Salton, Yang, and Yu, 1975). Probabilistic weighting schemes are discussed in more detail by Van Rijsbergen (1979). An early implementation of probabilistic retrieval for a practical (museum) retrieval system was

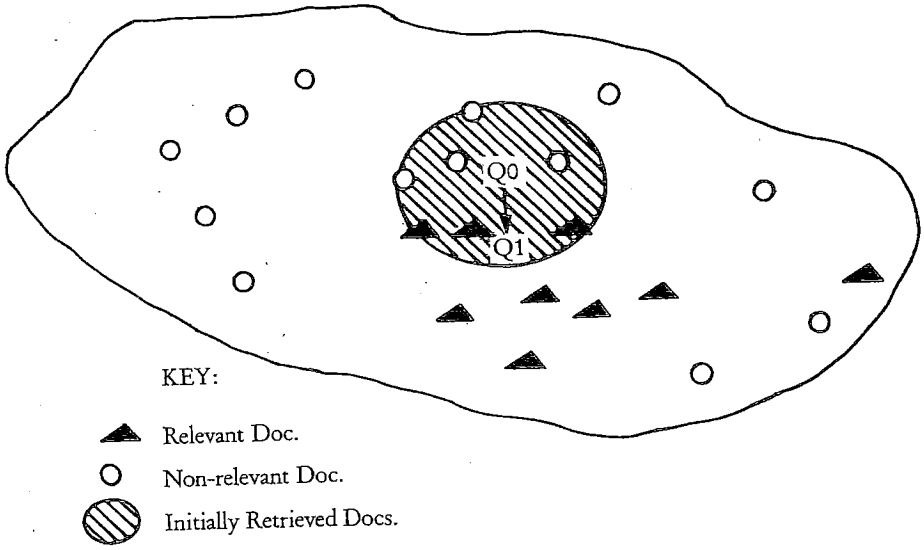

Figure 14. Vector feedback

done by Porter (1982). Probabilistic feedback has been incorporated by Buckley (1985) in recent versions of the SMART system and has also been adapted for several p-norm algorithms by this author. It is possible to obtain some of the benefits of probabilistic retrieval when accessing a conventional Boolean system, if sufficient processing is carried out by an intelligent front-end system such as that of Morrissey (1983).

In addition to supporting vector retrieval, the document-term matrix has been used to rationalize term or document clustering. The earliest thorough term-classification study, conducted by Sparck Jones in 1971, demonstrated that terms which co-occur in similar sets of documents tend to be ones that should be grouped into (thesaurus) classes. This technique could aid in semiautomatic selection of term classes when collections are not too large to warrant the relatively expensive processing.

Document clustering has been more thoroughly studied and shows greater promise. Figure 15 illustrates the process, where a number of similar documents are described by a centroid (named since the vector centroid can be so employed), where similar centroids are grouped under super centroids, etc. Since documents may often be about several topics, it is possible for the clustering to be overlapping, as shown for the second and fifth documents. A top-down search allows rapid identification of a few relevant documents, if at each point a depth-first selection is made of the best centroid to consider next.

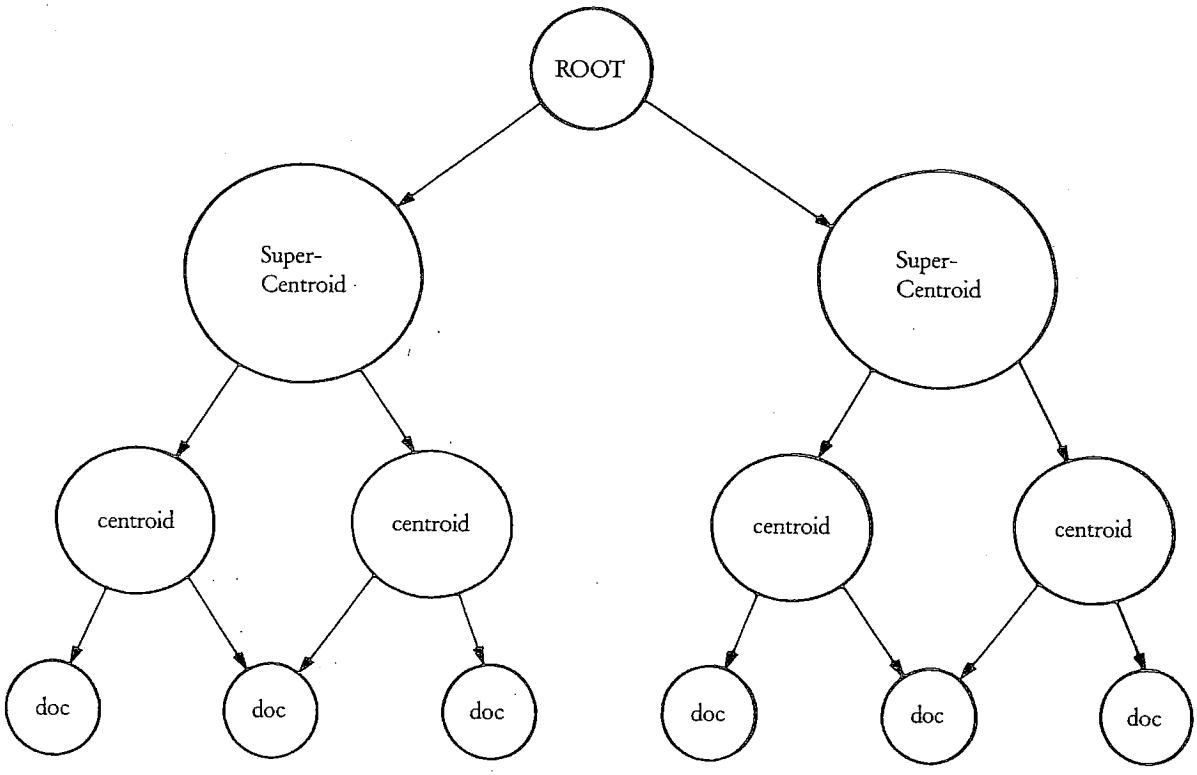

Figure 15. Cluster hierarchy organization

Clustering algorithms, such as those given by Hartigan (1975), have long been used in taxonomic analysis. Clustering using the well known single-link method to build searchable hierarchies was explored by Jardine and Van Rijsbergen (1971). For large collections, faster methods which still yield hierarchies that can be searched were developed by Dattola (1971) and Williamson (1974). Since cluster centroids are often quite large, top-down searching may be very expensive, so the bottom-up or hybrid scheme shown in Figure 16 was investigated by Croft (1978). A recent thorough examination of various clustering approaches by Voorhees (1985) highlights the complexity of the situation, identifies those situations where clustered searches may be better than inverted file searches, and encourages use of cluster connections to support better browsing systems.

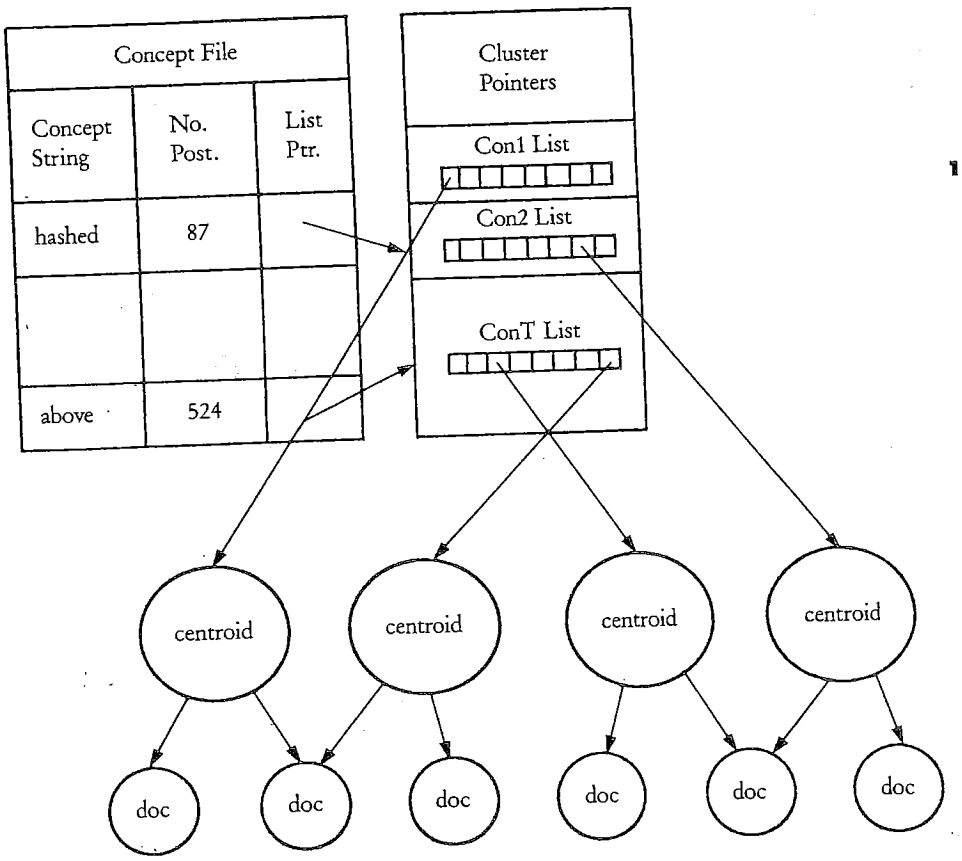

Figure 16. Hybrid inverted and clustered organization

Vector, probabilistic, and p-norm schemes all presume that the document collection has been automatically indexed. Figure 17 shows the key steps involved, for a hypothetical collection, on an example query. Function words like *with* and collection-specific high-frequency terms like *system* are stored in a stop word list. Incoming text is broken into tokens such as words, and a TRIE or hashing scheme is employed to identify stop words. The remaining terms are then alphabetized and repetitions avoided by the inclusion of frequency counts. Normal words are stemmed by a technique such as that of Lovins (1968). The system dictionary is then searched; new terms in documents are added when they first occur. At this time document frequency statistics can be updated. Next, a list of concept numbers

is constructed. Terms are weighted according to a scheme such as TF × IDF (described earlier), and vectors are built. Higher performance is possible if phrases are identified as they occur, and if terms with very low frequency are treated as markers of thesaurus classes, prior to weighting.

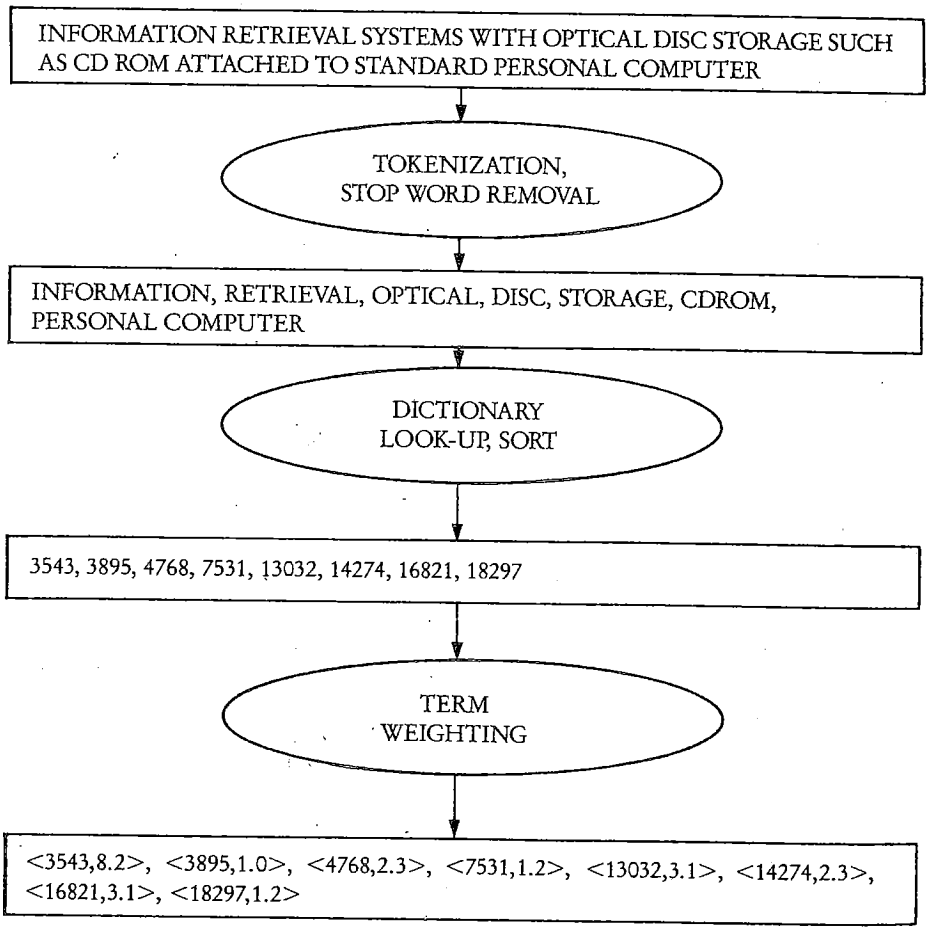

Figure 17. Simple example of automatic indexing

Given such an automatic indexing process, the overall process of indexing and vector or probabilistic retrieval can be seen in Figure 18.

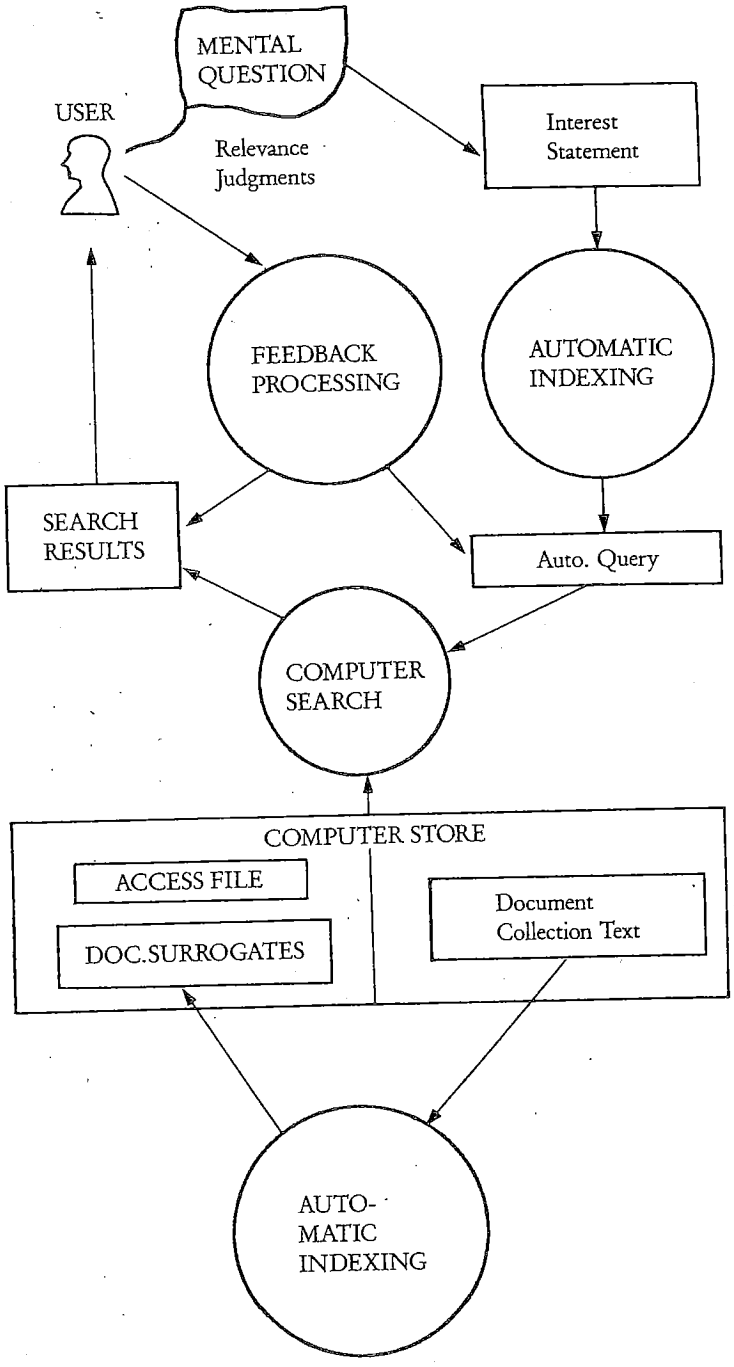

Figure 18. Automatic indexing and feedback retrieval

AI Models

The field of artificial intelligence has many subareas related to IR. Indeed, a number of IR researchers have worked on problems similar to those studied under AI. For example, O'Connor (1980) has employed linguistic methods and discourse analysis concepts, along with suitable heuristics, in order to select answer passages from texts. Oddy (1977) has focused on the user's model of an information need, and the specification of that need through a dialog with the computer, as the first and crucial step in retrieval. Hahn and Reimer (1984) use word expert parsing routines and discourse models in order to build a knowledge base that condenses texts so they can be more easily browsed and searched. Expert systems have likewise been employed by students like Yip (1979) to extend the work of Marcus on automating the job of search intermediaries.

At the most fundamental level, AI and IR are both concerned with the analysis and representation of the information content in texts. Thus, the early work by Evens on lexicons was in the computational linguistics area of AI but has since shown closer ties to IR. Related work by this author, for example, has demonstrated the value of relational models of the lexicon to aid in IR thesaurus class utilization. Earlier work on machine-readable dictionaries, such as by Amsler (1984), is now being extended to allow several full dictionaries to be automatically analyzed so that a comprehensive lexicon can be built to support work in both IR and AI.

Knowledge representation is another area of AI that closely relates to IR. Indexing methods and models of retrieval both depend on how the information content in texts can be analyzed and recorded. A taxonomy of representation schemes used in IR, formulated by Smith and Warner (1984), makes this point clearer and highlights the need for more IR work in document analysis. The work of Kimura (1984), in the domain of text processing, is of particular interest in providing a framework for modeling large documents.

During the last decade, the frame model of Minsky (1975) has served as the basis for many AI knowledge representation efforts. Frames support higher-level views of information complexes than the semantic networks which have also been intensively studied by AI researchers such as Ritchie and Hanna (1984). Similar in form, but more closely tied to the type of cognitive science work that has evolved at Yale, are scripts and more recent constructs for modeling memory. AI methods for handling temporal data have proved useful for certain IR applications such as the Reseda project Zarri, 1983, and further research is underway.

The larger question of how to analyze documents for IR is allied to AI work in natural language processing. Logic programming has been developed to help speed up development of language understanding and other tasks, as evidenced by the success of Prolog analysis systems by Pereira (1983). One version of Prolog has been used by Correira (1980) for computational analysis of stories. Simple but powerful routines for searching small document collections, for building schema representations of texts, and for question answering have also been demonstrated by Simmons (1984).

A Boolean query . . . will often evolve in steps; and although the final query will probably retrieve a reasonable number of hits, it may have a very complex syntactic and bizarre semantic structure.

A great deal of further work is needed in this area. Parsing systems must become more robust, to handle ungrammatical constructs and a variety of styles and genres identified by Hayes (1981) and others. Much of the progress to date relies on utilizing systems, like that of Sager (1975), that have been tailored for a particular sublanguage. Less detailed analysis is possible, as in DeJong's (1982) FRUMP, but requires a knowledge base for whatever subject areas are to be found in the text collection. The problem studied by Charniak (1982), of recognizing the context for a discussion so that appropriate specialized knowledge bases can be accessed, is unfortunately very hard to solve. Building schemas to represent story structure, investigated by Rumelhart (1975), is also difficult. Clearly, developing systems which Riesbeck (1982) hopes can realistically comprehend natural language texts, is very much a research problem.

At an even higher level, issues of matching user needs with texts are common to both IR and cognitive science studies. Designing a good interface to support human-computer interaction for IR will require extensive research and psychological testing beyond the recent work by Borgman (1984) and others. The need is even greater if simple Boolean systems are replaced by more sophisticated designs incorporating browsing, vector, probabilistic, and AI approaches. One of many AI efforts to develop such natural language dialog systems, with limited modeling of user knowledge and plans and with special domain knowledge, is the UC project of Wilensky et al. (1984), building a UNIX consultant.

The recent emphasis on constructing expert systems, surveyed by Hayes-Roth et al. (1983), is another AI trend that promises to be helpful in retrieval. Several efforts aim to capture the expertise of search intermediaries in dealing with many different information retrieval services and in constructing Boolean queries. A more ambitious effort is Thompson and Croft's (1985) integration of browsing with automatic retrieval, where the system has limited models of the users and has special knowledge about the appropriateness of a given retrieval model and search strategy in each situation.

CODER: Integration of Models

The CODER (Composite Document Effective/Extended/Expert Retrieval) project is a comprehensive effort by this author to bring together the best aspects of browsing, Boolean, p-norm, vector, probabilistic, and AI models into a new-generation retrieval system. The system is being developed partially in C, for special control and interface portions, but mostly in Prolog. Specifically, Naish's (1984) MU-Prolog is being employed since it includes improvements to avoid some of the failings of standard Prolog systems, and since it directly supports retrieval from large fact bases. The CODER effort includes a number of important components:

1. Constructing a comprehensive lexicon, by automatically analyzing two full English dictionaries.
2. Developing an architecture based on the blackboard model pioneered by Erman et al. (1980) in the Hearsay-II project.
3. Having multiple experts, each with private rule bases, as well as blackboard-based strategists for both analysis and access of documents, that can run as separate processes on multiple machines.
4. Focusing on sophisticated analysis of complex heterogeneous documents, to determine the type and structure of each composite document.
5. Automatically generating a knowledge representation for documents, including multiple vectors, frames, and relations.
6. Storing and updating models for each user, allowing human-computer interaction via a number of different devices (including simple or sophisticated graphics devices, and eventually, speech output).
7. Planning appropriate methods for both analysis and access, based on hypotheses posited by the many different experts.
8. Using AI, p-norm, vector, and probabilistic query techniques along with a knowledge base, inverted file, and cluster hierarchy to support searches, feedback, and browsing.

It is hoped that as the CODER system evolves, it will support a variety of different types of composite documents and serve the needs of a number of user classes, as shown in Figure 19.

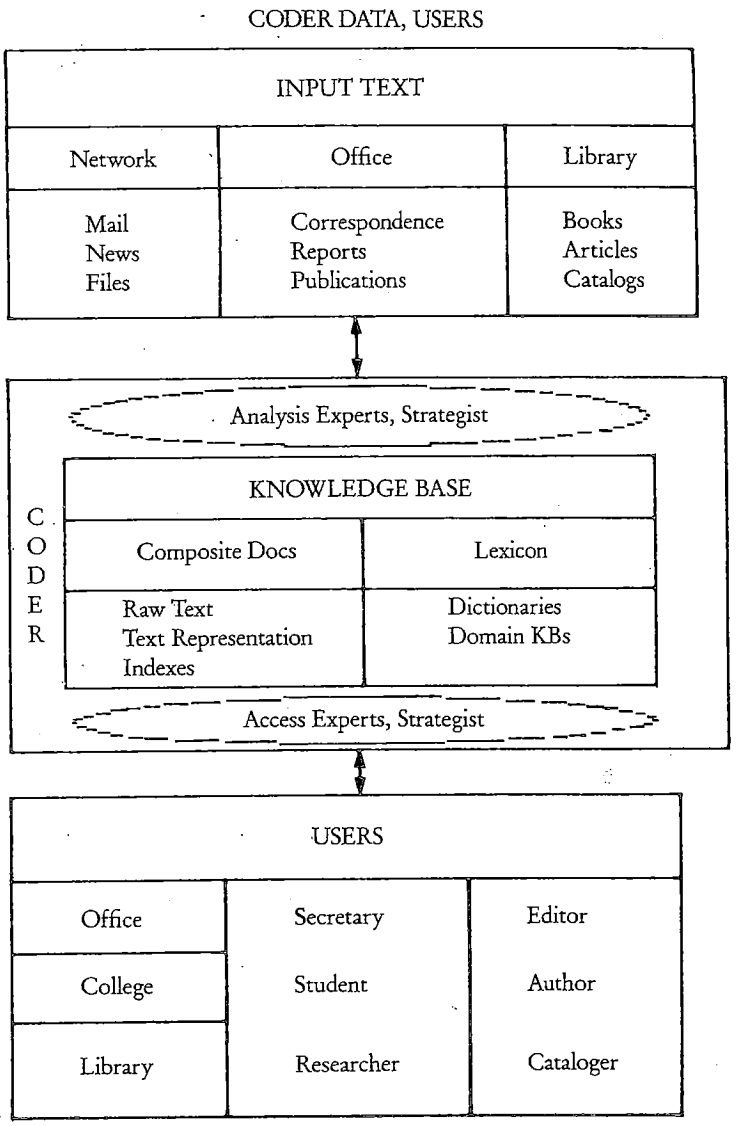

Figure 19. CODER

Past, Present, Future

Online Systems to Date

Today, the normal mode of access to bibliographic or full-text databases is through the use of online services. Large packet-switching public networks connect users to information retrieval services such as BRS, DIALOG, Dow Jones News/Retrieval, or MEDLARS. Information suppliers obtain their data from information providers.

Some of the databases available have been indexed, and a search often requires the aid of a trained intermediary to understand specialized thesauruses or other unique features. Boolean queries must be submitted; there is no ability to incorporate relative weights on terms, to use simpler or more powerful notations, to perform feedback operations, to get ranking of output, to rapidly browse, and so on. The investment in current retrieval software and emphasis on providing fast and low-cost services has made such enhancements irrelevant to managers of most online services.

CD ROM Systems Being Developed

At the most fundamental level, AI and IR are both concerned with the analysis and representation of the information content in texts.

With the advent of CD ROM systems, the initial focus has been to simply make the power of a large online service directly available to the user of a PC. Microcomputer versions of systems like BRS and BASIS provide essentially the same functionality as the mainframe software, with an almost identical user interface. In some cases, though, menu rather than command languages are employed.

Development work has focused on building suitable device drivers, designing file systems, compressing or replicating data, helping information providers organize their data for mastering, and struggling to achieve rapid response to commands. Since many PCs support graphics or color displays, some attention has also been paid to improving the simple line-by-line interaction mode provided by online services.

Another concern has been the integration of CD ROM with online systems, so that when updates become available they can be immediately accessed. Online searching, downloading to local magnetic disks, or distribution of diskettes or laser cards are all possible solutions.

Some more innovative developments are also taking place. Grolier has pioneered work on electronic encyclopedias to be readily accessible through PCs with CD ROM readers. The SIRE system (mentioned earlier) has been adapted to run on PCs as well as large minicomputers, and will support CD ROM databases. In both of these cases, more flexible interfaces than usual are provided.

Future Possibilities

The challenge raised in this chapter is to make a quantum leap forward in the practice of information retrieval by integrating the advances in processor and storage technology with the fruits of retrieval research. Storage methods have long been of interest to IR investigators such as Goldstein (1984), and today, with optical discs like CD ROMs becoming readily

available, are even more exciting. Studying such large databases will require new experimental approaches, such as the simulation work of Tague (1981) et al. which relies on studies of the distributions found in existing collections. But it is likely that many research methods can be applied without a great deal of further adaptation.

Fully automatic processing . . . will transform the new papyrus, CD ROM, into a portal to knowledge . . .

What is perhaps most important is that with the advent of hardware that can store and process very large text collections, and is inexpensive enough so as to be targeted for the home market, there will be a new emphasis on effective as well as efficient retrieval. Fully automatic processing, where a natural language or an enhanced Boolean query can be handled, and where hidden but sophisticated analysis, indexing, search, and presentation capabilities are provided, will transform the new papyrus, CD ROM, into a portal to knowledge much like the memex proposed by Vannevar Bush in 1945!

About the Author

Edward A. Fox is assistant professor of computer science at Virginia Tech. He is coeditor of *ACM SIGIR Forum* and editor of *IRList Digest,* an electronic newsletter; ARPANET, BITNET, and CSNET. He has a B.S. in electrical engineering from MIT and M.S./Ph.d. in computer science from Cornell. He is a member of ACL, ACM, AFIS, and Sigma Xi. Current research in information retrieval utilizes UNIX, C, Prolog, microcomputers, computational linguistics, and expert systems.

Virginia Technical University
Department of Computer Science
562 McBride Hall
Blacksburg, VA 24061

References

Amsler, R. A. 1984. Machine-readable dictionaries. *ARIST,* 19:161-209.

Attar, R., and A. S. Fraenkel. 1977. Local feedback in full-text retrieval systems. *J. ACM,* July, pp. 397-417.

Bayer, R., and E. McCreight. 1972. Organization and maintenance of large ordered indexes. *Acta Informatica,* 1(3):173-189.

Bichteler, J. and E. A. Eaton III. 1980. The combined use of bibliographic coupling and cocitation for document retrieval. *J. Am. Soc. Inf. Sci.,* July, 31(4).

Blair, D. C. and M. E. Maron. 1985. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Commun. ACM*, March, 28(3):289-299.

Bookstein, A. 1980. Fuzzy requests: an approach to weighted Boolean searches. *J. Am. Soc. Inf. Sci.*, July, 31(4):240-247.

Borgman, Christine L. 1984. Psychological research in human-computer interaction. *ARIST*, 19:33-64.

Borko, H. and C. L. Bernier. 1978. *Indexing Concepts and Methods*. New York:Academic Press.

Boyer, R. S. and J. S. Moore. 1977. A fast string searching algorithm. *Comm. ACM*, October, 20(10):762-772.

Buckley, C. 1985. *Implementation of the SMART Information Retrieval System*. TR 85-686, May, Cornell Univ., Dept. of Comp. Sci.

Bush, V. 1945. As we may think. *Atlantic Monthly*, July, 176:101-108.

Charniak, E. 1982. Context recognition in language comprehension. In *Strategies for Natural Language Processing*, ed. by Wendy G. Lehnert and Martin H. Ringle, Hillsdale, N.J.: Lawrence Erlbaum Assoc., pp. 435-454.

Choeka, Y. 1980. Computerized full-text retrieval systems and research in the humanities: the Responsa Project. *Computers and the Humanities*, Nov., 14(3): 153-169.

Cleverdon, C. W., and E. M. Keen. 1968. Factors determining the performance of indexing systems. Cranfield, England: *Aslib Cranfield Research Project*, Vol. 1 and 2.

Comer, D. 1979. The ubiquitous B-tree. *ACM Comp. Surveys*, June, 11(2):121-137.

Cook, P. R. 1984. Electronic encyclopedias. *Byte*, July, 9(7): 151-170.

Correira, A. 1980. Computing story trees. *Amer. J. Comp. Ling.*, 6(3-4): 135-149.

Crawford, R. G. 1981. The relational model in information retrieval. *J. Am. Soc. Inf. Sci.*, 32(1): 51-64.

Croft, W. B. 1978. Organizing and Searching Large Files of Document Descriptions. Dissertation. Cambridge Univ., England.

Croft, W. B., and M. T. Pezarro. 1982. Text retrieval techniques for the automated office. In *Office Information Systems*, ed. by N. Naffah, North-Holland, Amsterdam, pp. 565-576.

Dattola, R. T. 1971. Experiments with a fast algorithm for automatic classification. In *The SMART Retrieval System, Experiments in Automatic Document Processing*, ed. by G. Salton, Prentice Hall, Englewood Cliffs, N.J.

Dattola, R. T. 1979. FIRST: Flexible Information Retrieval for Text. *J. Am. Soc. Inf. Sci.*, 30(1):9-14.

DeJong, G. 1982. An overview of the FRUMP system. In *Strategies for Natural Language Processing*, ed. by Wendy G. Lehnert and Martin H. Ringle, Hillsdale N.J.: Lawrence Erlbaum Assoc., pp. 149-176.

Erman, L. D., F. Hayes-Roth, V. R. Lesser, and D. R. Reddy. 1980. The Hearsay-II Speech-Understanding System: integrating knowledge to resolve uncertainty. *ACM Comp. Surveys*, 12: 213-253.

Evens, M. W., and R. N. Smith. 1979. A lexicon for a computer question-answering system. *Am. J. Comp. Ling.*, Microfiche 83.

Faloutsos, C. 1985. Access methods for text. *ACM Comp. Surveys*, March, 17(1): 49-74.

Fox, E. A. 1983. Combining information in an extended automatic information retrieval system for agriculture. *Infrastructure of an Information Society* (Proc. 1st Int. Info. Conf. Egypt, 13-16 Dec. 1982), North-Holland, Amsterdam.

Fox, E. A. 1983. Extending the Boolean and vector space models of information retrieval with p-norm queries and multiple concept types. Dissertation, Aug., Cornell University, University Microfilms Int., Ann Arbor, Mich.

Fox, E. A. 1983. *Some Considerations for Implementing the SMART Information Retrieval System under UNIX*. TR 83-560, Sept., Cornell Univ., Dept. of Comp. Sci.

Fox, E. A. 1983. *Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts*. TR 83-561, Sept., Cornell Univ., Dept. of Comp. Sci.

Fox, E. A. 1984. *Improved Retrieval Using a Relational Thesaurus for Automatic Expansion of Boolean Logic Queries*. Proc. Workshop on Relational Models of the Lexicon, June 29, Stanford Univ.

Fox, E. A. 1985. Composite Document Extended Retrieval: An Overview. In *Res. & Dev. in Inf. Ret.*, Eighth Annual Int. ACM SIGIR Conf., Montreal, June 5-7, pp. 42-53.

Frei, H. P. and J. F. Jauslin. 1983. Graphical presentation of information and services: a user oriented interface. *Inf. Tech.: Res. Dev.*, Jan., 2(1): 23-42.

Frei, H. P. and J. F. Jauslin. 1984. Two-dimensional representation of information retrieval services. In *Representation and Exchange of Knowledge as a Basis of Information Processes*, ed. by Hans J. Dietschmann, North-Holland, New York: pp. 383-396.

Fujitani, L. 1984. Laser optical disk: the coming revolution in on-line storage. *Commun. ACM*, June, 27(6): 546-554.

Goldberg, A., D. Robson, and D. H. H. Ingalls. 1982. *Smalltalk-80: the language and its implementation*, Addison-Wesley, Menlo Park, Calif.

Goldberg, A., D. Robson, and D. H. H. Ingalls. 1982. *Smalltalk-80: the interactive programming environment*, Addison-Wesley, Menlo Park, Calif.

Goldstein, C. M. 1984. Computer-based information storage technologies. *ARIST*, 19: 65-96.

Hahn, U., and U. Reimer. 1984. Heuristic text parsing in 'topic': methodological issues in a knowledge-based text condensation system. In *Representation and Exchange of Knowledge as a Basis of Information Processes*, ed. by Hans J. Dietschmann, North-Holland, New York, pp. 143-163.

Hall, P. A. V., and G. R. Dowling. 1980. Approximate string matching. *ACM Comp. Surveys*, 12(4): 381-402.

Hartigan, J. A. 1975. *Clustering Algorithms*. John Wiley and Sons, New York.

Haskin, R. 1980. Hardware for searching very large text databases. In *Proc. 5th Workshop on Comp. Arch. for Non-Numeric Proc., ACM, March, New York, pp. 49-56.*

Hayes, P., and G. V. Mouradian. 1981. Flexible parsing. *Amer. J. Comp. Ling.*, 7(4): 232-242.

Hayes-Roth, F., D. A. Waterman, and D. B. Lenat, eds. 1983. *Building Expert Systems*, Addison-Wesley, Reading, Mass.

Heap, H. S. 1978. *Information Retrieval, Computational and Theoretical Aspects*. Academic Press, New York.

Hollaar, L. A. 1985. A testbed for information retrieval research: the Utah Retrieval System Architecture. *Proc. 8th Annual. Int. ACM SIGIR Conf. on R&D in Inf. Ret.*, June 5-7, Montreal, pp. 227-232.

Jardine, N. and C. J. Van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. *Inf. Stor. and Ret.*, Dec., 7(5): 217-240.

Jennings, M. 1984. The electronic manuscript project. *Bulletin Am. Soc. Inf. Sci.*, Feb. 9, 10(3): 11-13.

Katzer, J., et. al. 1982. *A Study of the Overlap Among Document Representations*. Syracuse Univ. Sch. of Info. Studies.

Kimura, G. D. 1984. *A Structure Editor and Model for Abstract Document Objects*. Dissertation. Tech. Report No. 84-07-04, July, Dept. of Comp. Sci., Univ. Washington.

Lancaster, F. W. 1978. *Toward Paperless Information Systems*, New York: Academic Press.

Lehnert, W. G. 1978. *The Process of Question Answering: A Computer Simulation of Cognition*. Hillsdale, N.J.: Lawrence Erlbaum Assoc.

Lovins, B. J. 1968. Development of a stemming algorithm. *Mech. Trans. and Comp. Ling.*, March-June, 11(1-2): 11-31.

Macleod, I. A. and R. G. Crawford. 1983. Document retrieval as a database application. *Inf. Tech.: Res. Dev. Applications*, Jan., 2(1): 43-60.

McGill, M. J., et al. 1976. *Syracuse Information Retrieval Experiment (SIRE): Design of an On-Line Bibliographic Retrieval System*. ACM SIGIR Forum, Spring, 10(4): 37-44.

McGill, M. J., M. Koll, and T. Noreault. 1979. *An Evaluation of Factors Affecting Document Ranking By Information Retrieval Systems*. Syracuse Univ. Sch. of Info. Studies.

Meadow, C. T., and P. A. Cochrane. 1981. *Basics of Online Searching*. New York: John Wiley and Sons.

Minsky, M. 1975. A framework for representing knowledge. In *The Psychology of Computer Vision*, ed. by P. Winston, New York: McGraw-Hill.

Morrissey, J. 1983. An intelligent terminal for implementing relevance feedback on large operational retrieval systems. In *Res. & Dev. in Inf. Ret., Proc., Berlin, May 18-20, 1982*, ed. by G. Salton and Hans-Jochen Schneider, Springer-Verlag, Berlin, .pp 38-50.

Naish, L. 1984. *MU-Prolog 3.1db Reference Manual*, Melbourne Univ.

Noreault, T., M. Koll and M. J. McGill. 1977. Automatic ranked output from Boolean searches in SIRE. *J. Am. Soc. Inf. Sci.*, Nov., 28(6):333-339.

O'Connor, J. 1980. Answer-passage retrieval by text searching. *J. Am. Soc. Inf. Sci.*, 31(4):227-239.

Oddy, R. N. 1977. Information retrieval through man-machine dialogue. *J. Doc.*, March, 33(1):1-14.

Paice, C. D. 1984. Soft evaluation of Boolean search queries in information retrieval. *Inf. Tech.: Res. Dev. Applications*, 3(1): 33-42.

Pereira, F. 1983. *Logic for Natural Language Analysis*, Tech. Note 275, Jan., SRI Int.

Pfaltz, J. L., W.H. Berman, and E.M. Cagley. 1980. Partial-match retrieval using indexed descriptor files. *Commun. ACM*, Sept., 23(9): 522-528.

Porter, M. F. 1982. Implementing a probabilistic information retrieval system. *Inf. Tech.: Res. Dev.*, 1(2).

Riesbeck, C. K. Realistic language comprehension. 1982. In *Strategies for Natural Language Processing*, ed. by Wendy G. Lehnert and Martin H. Ringle, Hillsdale, N.J.: Lawrence Erlbaum Assoc., pp. 435-454.

Ritchie, G. D. and F. K. Hanna. 1984. Semantic networks—a general definition and a survey. *Inf. Tech.: Res. Dev. Applications*, 3(1): 33-42.

Robertson, S. E. and K. Sparck Jones. 1976. Relevance weighting of search terms. *J. Am. Soc. Inf. Sci.*, 27(3): 129-146.

Rocchio, Jr., J. J. 1971. Relevance feedback in information retrieval. In *The SMART Retrieval System, Experiments in Automatic Document Processing*, ed. by G. Salton, Englewood Cliffs, N.J.: Prentice Hall.

Rumelhart, D. E. 1975. Notes on a schema for stories. In *Representation and Understanding*, ed. by D.G. Bobrow and A. Collins, New York: Academic Press, pp. 211-236.

Sager, N. 1975. Sublanguage grammars in science information processing. *J. Am. Soc. Inf. Sci.*, Jan.-Feb., 26(1): 10-16.

Salton, G. 1972. A new comparison between conventional indexing (Medlars) and text processing (SMART). *J. Am. Soc. Inf. Sci.*, 23(2): 75-84.

Salton, G., C. S. Yang, and C. T. Yu. 1975. A theory of term importance in automatic text analysis. *J. Am. Soc. Inf. Sci.*, Jan.-Feb., 26(1): 33-44.

Salton, G., A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing, *Commun. ACM*, Nov., 18(11):613-620.

Salton, G. 1980. The SMART System 1961-1976: experiments in dynamic document processing. In *Encyclopedia of Library and Information Science*, pp. 1-36.

Salton, G. and M. J. McGill. 1983. *Introduction to Modern Information Retrieval*, New York: McGraw-Hill.

Salton, G., C. Buckley, and E. A. Fox. 1983. Automatic query formulations in information retrieval. *J. Am. Soc. Inf. Sci.*, July, 34(4): 262-280.

Salton, G., E. A. Fox, and H. Wu. 1983. Extended Boolean information retrieval. *Commun. ACM*, Nov., 26(11): 1022-1036.

Salton, G., E. A. Fox, and E. Voorhees. 1985. Advanced feedback methods in information retrieval, *J. Am. Soc. Inf. Sci.*, 36(3): 200-210.

Schank, R. C. and R. P. Abelson. 1977. *Scripts, Plans, Goals and Understanding*, Hillsdale, N.J.: Lawrence Erlbaum Assoc.

Simmons, R. F. 1984. *Computations from the English*, Englewood Cliffs, N.J.: Prentice-Hall.

Small, H. 1980. Cocitation context analysis and the structure of paradigms. *J. Doc.*, Sept., 36(3): 183-196.

Smith, L. C. and Warner, A. J. 1984. A taxonomy of representations in information retrieval system design. In *Representation and Exchange of Knowledge as a Basis of Information Processes*, ed. by Hans J. Dietschmann, North-Holland, New York, pp. 31-49.

Sparck Jones, K. 1971. *Automatic Keyword Classifications*, London: Butterworths.

Sparck Jones, K. and C. J. Van Rijsbergen. 1976. Information retrieval test collections. *J. Doc.*, March, 30(4).

Stonebraker, M., et al. 1983. Document processing in a relational database system. *ACM Trans. on Office Inf. Systems*, April, 1(2).

Tague, J., M. Nelson, and H. Wu. 1981. Problems in the simulation of bibliographic retrieval systems. In *Information Retrieval Research*, ed. by R. NM. Oddy, S. E. Robertson, C. J. Van Rijsbergen, and P. W. Williams, London: Butterworths, pp. 236-255.

Tenopir, Carol. 1984. Full-text databases. *ARIST*, 19: 215-246.

Thompson, R. H., and W. B. Croft. 1985. An expert system for document retrieval. *Proc. Expert Systems in Gov. Symp.*, IEEE, Oct., pp. 448-456.

Van Rijsbergen, C. J. 1979. *Information Retrieval: Second Edition*. London: Butterworths.

Voorhees, E. M. 1985. *The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval.* Dissertation. TR 85-705, Oct., Cornell Univ., Dept.of Comp. Sci.

Weyer, S. A. 1982. *Searching for Information in a Dynamic Book*. SCG-82-1. Feb., Xerox PARC, Palo Alto, Calif.

Wilensky, R., Y. Arens, and D. Chin. 1984. Talking to UNIX in English: an overview of UC. *Commun. ACM*, 27(6): 574-593.

Williamson, R. E. 1974. *Real-Time Document Retrieval.* Dissertation. June, Cornell Univ.

Yip, Man-Kam. 1979. *An Expert System for Document Retrieval.* M. S. Thesis. M. I. T., Cambridge, Mass.

Zarri, G. P. 1983. An outline of the representation and use of temporal data in the RESEDA system. *Inf. Tech.: Res. Dev. Applications*, July.