A Call for Integrating Advanced Information Retrieval Models with CD-ROM / Microcomputer Systems

by
Edward A. Fox

June 1986

TR-86-14

A Call for Integrating Advanced Information Retrieval Models with CD-ROM / Microcomputer Systems†

Edward A. Fox
Department of Computer Science
Virginia Tech
Blacksburg VA 24061

ABSTRACT

Recent advances in computer hardware and storage devices will allow inexpensive personal systems to be used by individuals to rapidly access vast collections of text. Research into database management, artificial intelligence, and information retrieval can all be applied to develop advanced retrieval systems. Retrieval models based on browsing, extended Boolean, vector, probabilistic, and artificial intelligence approaches have all been advanced as more effective for searchers than conventional methods. The CODER project aims to integrate these techniques. Ultimately it is hoped that CD-ROM based information retrieval systems will be released with many of the capabilities mentioned.

CR Categories and Subject Descriptors: H.3 **[Information Storage and Retrieval]**; I.2.1 **[Artificial Intelligence]:** Applications and Expert Systems; I.2.4 **[Artificial Intelligence]:** Knowledge Representation Formalisms and Networks; I.2.7 **[Artificial Intelligence]:** Natural Language Processing

General Terms: Design, Experimentation

Additional Keywords and Phrases: CD-ROM, CDROM, browsing, Boolean queries, p-norm queries, extended Boolean logics, clustering, inverted files, fuzzy sets, feedback, automatic indexing, CODER project

---

† The following report was previously printed, in modified form, as:
"Information Retrieval: Research into New Capabilities," in *CD ROM. The New Papyrus*, eds. Lambert, Steve and Ropiequet, Suzanne, Microsoft Press (1986), pp. 143-74.

1 INTRODUCTION

1.1 Background

The long anticipated day of personal access to large databases is nearly here! Will the research methods developed in the field of information retrieval over the last four decades be useful? Will they be considered by developers of PC-based CD-ROM systems? This report is an attempt to answer these two pressing questions, by highlighting areas where "yes" is the proper response.

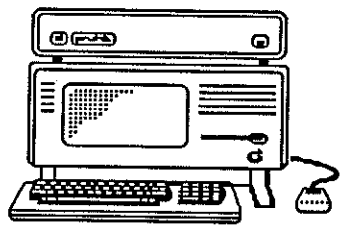
In 1945, Vannevar Bush published his seminal article, "As We May Think" [10], in which he envisoned a device called Memex that would transform a desk into a portal to the world's stored knowledge. Not only would it be possible to access information, but it would also be possible to record one's trail of investigations and share those associations with others.

In the last four decades, researchers in the little known field "information retrieval" have struggled with technological and scientific issues in their attempt to create systems with some of the capabilities of Memex. While Bush's dream was initially described in terms of microfilm and mechanical devices, today's designers have microcomputers, fiber optic networks, and laser discs. Technologically, we are much closer to the "paperless society" [50]. 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 because of the tools available. 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. At Virginia Tech, for example, freshman majoring in computer science in 1985 bought microcomputers as shown in Figure 1, with 1 megabyte of RAM, 10 megabytes of disk, a 400,000 byte diskette drive, a mouse, and a fairly high-resolution bit-mapped display. They use the sophisticated UNIX (Trademark of AT&T Bell

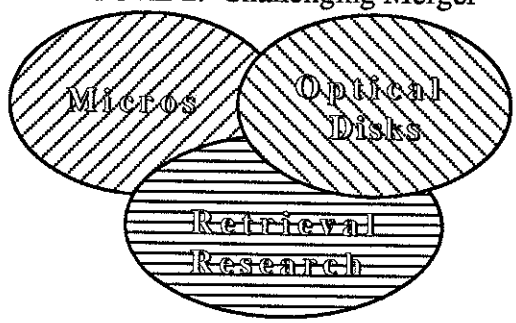
Laboratories) operating system, and have compilers for the C, FORTRAN, and PASCAL languages. In 1986 they will each be given a powerful automatic information retrieval package, descended from the SMART system [77], which in 1978 required a large IBM 370 computer to function.

FIGURE 1: Virginia Tech CS Micro



Microcomputers and laser discs have captured the imagination of computer scientists and users alike [34]. What is needed is a coalescing of those two developments with the fruits of retrieval research, as shown in Figure 2. Now it is possible to leap-frog beyond current systems with quantum improvements in all three areas. The result will be an exciting range of products bringing us another step closer to what Vannevar Bush really envisioned.

FIGURE 2. Challenging Merger



1.2 Focus

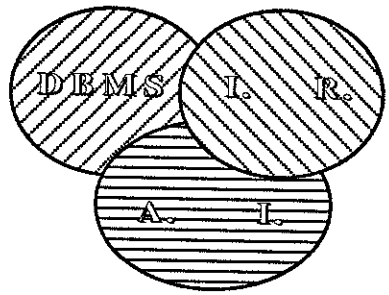
Papyrus was the world's first paper, and revolutionized the recording of information. Today, CD-ROM is playing a similar role. However, providing access to that information becomes of even more paramount importance, due to the sheer volume of material available. Access to information has many dimensions, though, as can be seen in a recent survey [25].

This report will focus primarily on access methods and techniques developed in the area of information retrieval. That itself is a broad subject; more detailed texts such as [44], [92], or [78] should be consulted by readers interested in learning more. Because of its focus on research, this report will not describe current systems, services, or products except for purposes of contrast or explanation. Rather, discussion will center on models, on experiments testing system effectiveness, and on indexing/retrieval/interface methods.

2 RESEARCH ISSUES

Information retrieval (IR) is a cross-disciplinary field investigated by librarians, linguists, psychologists, and computer or information scientists. As can be seen in Figure 3, two of the closest areas to IR are the study of database management systems (DBMS), and of artificial intelligence (AI).

FIGURE 3. Related Fields



What is distinctive about IR is the need to work with:

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

Certainly, DBMS and AI researchers confront some of these issues, but they are not necessarily of central interest. Point 2 above is clearly important for linguists; point 3 for library and information scientists; and point 4 for cognitive psychologists. Each of these points will be touched on in later sections, while a slightly different topical organization will be considered below.

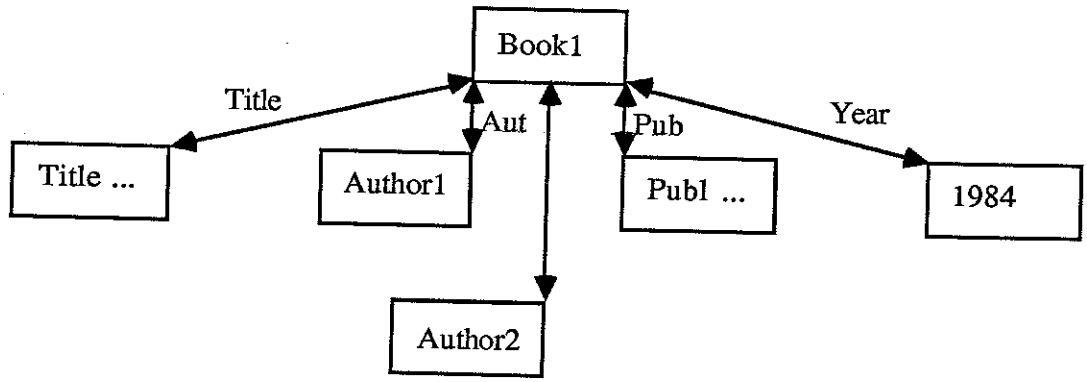
2.1 Representation

How information is viewed by system designers and users is one of the key differences between the areas of DBMS, IR, and AI. Figure 4 illustrates the distinction in terms of a collection of bibliographic citations. The tabular form shown is similar to that possible with relational database management systems; since the reader is probably familiar with this viewpoint, little further discussion below is warranted. The text form, with embedded codes, is used as input to several information retrieval systems. The network diagram at the bottom is illustrative of the various AI schemes for semantic network types of knowledge representation.

FIGURE 4. Representations Of Citation Data In Database, Ir, Ai Systems

BIBLIOGRAPHIC CITATIONS			
TITLE	AUTHOR	PUBL.	YEAR
Title1	Author1	New P.	1984
Title1	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
```



Due to the proliferation of DBMS packages, attention has been given to integrating IR and DMBS technology. Dattola developed the FIRST system merging a network database and a clustered retrieval system [20]. Crawford advocated the adaptation of the relational model for IR purposes [17], and with Macleod viewed IR as a database application [53]. Developers of database systems have made special adaptations to better support document processing [88]. Some

commercially available retrieval systems claim to integrate DBMS and IR processing, but no elegant merger of the concepts has yet emerged. Clearly many techniques employed in database systems, such as B-trees [3], are used in retrieval systems in various components [14] such as dictionaries, but direct use of a database for information retrieval is problematic [28]. Partial match methods suitable for DBMS [66] may also be unsuitable for IR applications when variable length queries and large collections are involved.

Other work has focused on the non-textual data relating to documents, and shown the value of that additional type of information. Citation data can lead to co-citation diagrams which highlight the areas in a field and inter-relationships involved [84]. Bibliographic coupling is also of value [4]. When these relationships are used in concert with descriptive and textual data, the resulting representation, according to results with two test collection developed to explore this possibility [29], can lead to even more effective retrieval [26].

2.2 Text Processing

Traditionally, text has appeared in manuscripts, journals, reference works, periodicals, reports, newspapers, correspondence, and other forms. Computer methods to aid in the creation and layout of text items, such as word processing, electronic messaging, and photocomposition, have led to the availability of machine readable texts in ever increasing numbers. Standards work will accentuate that trend [47]. Attendant issues of analyzing, indexing, storing, and accessing text have been central to much of the work in modern IR [78].

The simplest way to access text is to search it directly. Much as people scan the words on a page, much faster computers can also search sequentially. Clever algorithms can speed up the process [8], and further improvements are possible with special hardware [41]. Research in this area is ongoing at such sites as Utah [45]. When inexact string matching is needed, additional complexity results [39]. As is discussed later, however, more complex models of text documents can allow large collections to be efficiently searched with less expensive systems.

Relating to modeling document structure is the issue of indexing document content. Most indexing is performed by human beings without the aid of computers [7]. Early studies have

shown however, that simple indexing languages that can be processed on computers can be highly effective [13]. Indeed, when article titles and abstracts are automatically indexed, retrieval appears to be as good as when high quality manual indexing takes place [74]. With the advent of numerous large full-text databases [90], however, searches of unindexed "free-text" often take place instead. Since a majority of the relevant entries are likely to be missed in such searches [54], it may be more appropriate to employ automatic indexing techniques, especially in the many situations where manual indexing is impossible or awkward (eg., for a person's electronic mail, in an office [19], or in a dynamic collection). Complex morphological analysis and sophisticated retrieval methods incorporated into one full-text retrieval system suggest that automatic processing may indeed be of great value [12].

2.3 Experimental Collections

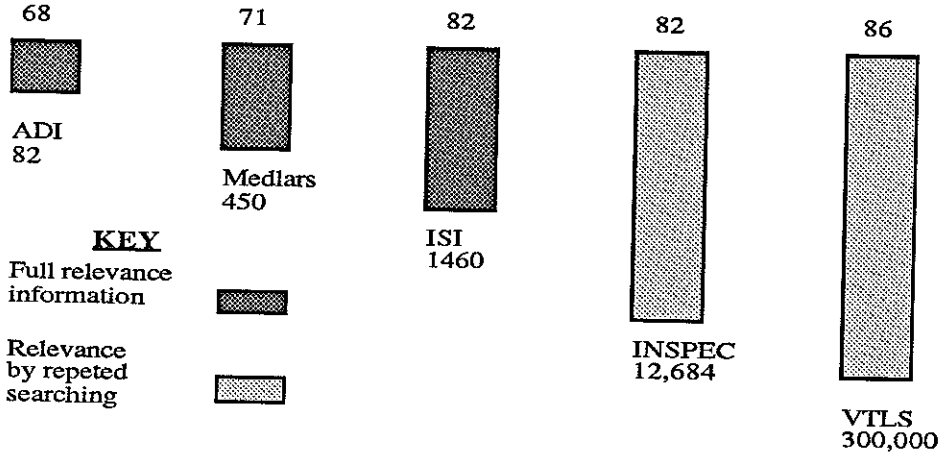
When people search through text collections, they are able to decide whether located items are relevant to their particular search need. Computerized retrieval systems should find exactly those items that are relevant -- ignoring items that are non-relevant, and locating all of the relevant set. That is to say, they should have high *precision* as well as high *recall*.

To determine the most effective indexing and retrieval methods, therefore, test collections have been devised, including: a set of "documents," a set of queries, and a list of which documents are relevant to each query. Ideally, there should be a large number of documents and queries so that statistically valid comparisons can be made, and so that the test collection is of sufficient size to be considered "realistic" [87].

Information retrieval experiments have often been viewed as irrelevant (eg., [54]) because of the use of small collections. However, this author has observed a number of results which do scale up from small to fairly large collections. In addition, bigger and bigger collections have been used over the years, as is shown in Figure 5. The first four collections listed were used by this author in previous studies [27], and the fifth should be used for some 1986 studies. For the larger collections, recall is difficult to measure since searchers will only supply relevance judgments on a

small subset of the documents. Nevertheless, various relative measures or estimation techniques can be employed.

FIGURE 5. Growth In Size Of Test Collections



It has been shown that when a number of searches are carried out, there is little overlap between the sets of documents retrieved [48]. This suggests that for high recall on large collections, multiple searches should be conducted. One approach would then be to carry out searches according to each of the models described below.

3 Models

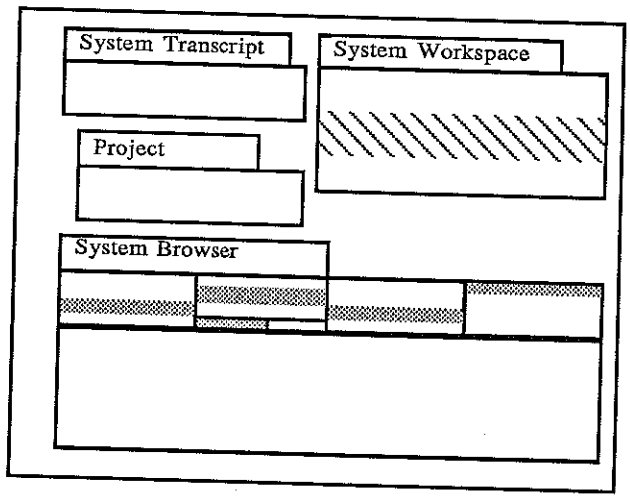
Much of the research in IR has been based on one of four different types of models of the field. Traditionally, the Boolean model has been most widely upheld. The vector and probabilistic models have shown promise in laboratory tests. With recent advances in display and pointing devices, browsing models 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 their complexity.

3.1 Browsing

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 is of particular interest [33].

The Smalltalk-80 system has been implemented to provide such an environment [36]. Since hierarchical organizations of knowledge are commonplace, the language and displays support that perspective [35]. An example of the "System Browser" is shown in Figure 6. Once Smalltalk is thoroughly understood, program development for broad classes of applications becomes especially convenient.

FIGURE 6. Smalltalk-80 with browser



Weyer explored the value of a browsing model of searching books [94]. Figure 7 gives a crude portrayal. 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, other pointers, or by the aid of text searching.

FIGURE 7. Dynamic Book

AREAS:

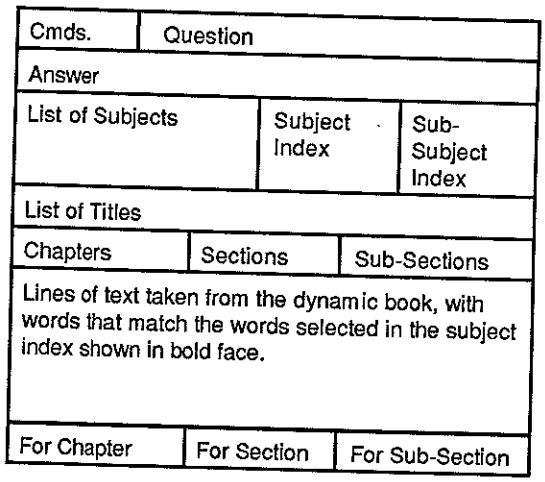
Command

Subject

Title

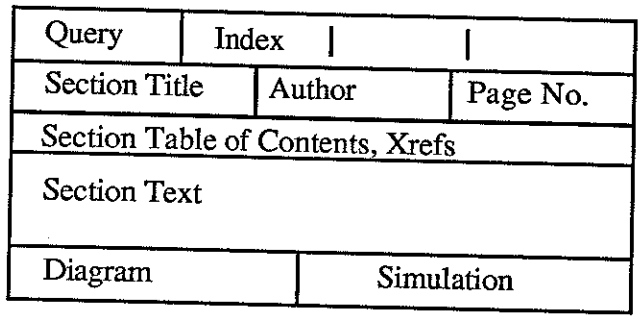
Text

Subject References



On a somewhat larger scale, electronic encyclopedias provide browsing-based access to large reference works [15]. As can be seen in Figure 8, all of the cross-references found in an encyclopedia can supplement what is available in a dynamic book. Furthermore, special capabilities can be build into such systems to understand units of measure, foreign words, or alternative map presentations. Lessons, simulations, and other enhancements can also be incorporated.

FIGURE 8. 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 [32]. It employs some of the most useful search techniques along with multi-window display and browsing. The appeal of this model has encouraged other researchers to incorporate some of the ideas of Caliban in more recent systems that employ AI methods as well [91].

3.2 Boolean and P-Norm

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 9 illustrates this approach. Part (a) is an English language statement of interest that might be appropriate for a reader of this report. Part (b) is a typical expression of such an interest as a Boolean query. For those with a computer science bent, part (c) gives an equivalent form using "prefix" notation. That can be directly mapped into the tree structure in part (d) which illustrates the relationship of the various concepts.

FIGURE 9. TRANSFORMATIONS TO BOOLEAN QUERY REPRESENTATIONS

A)

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

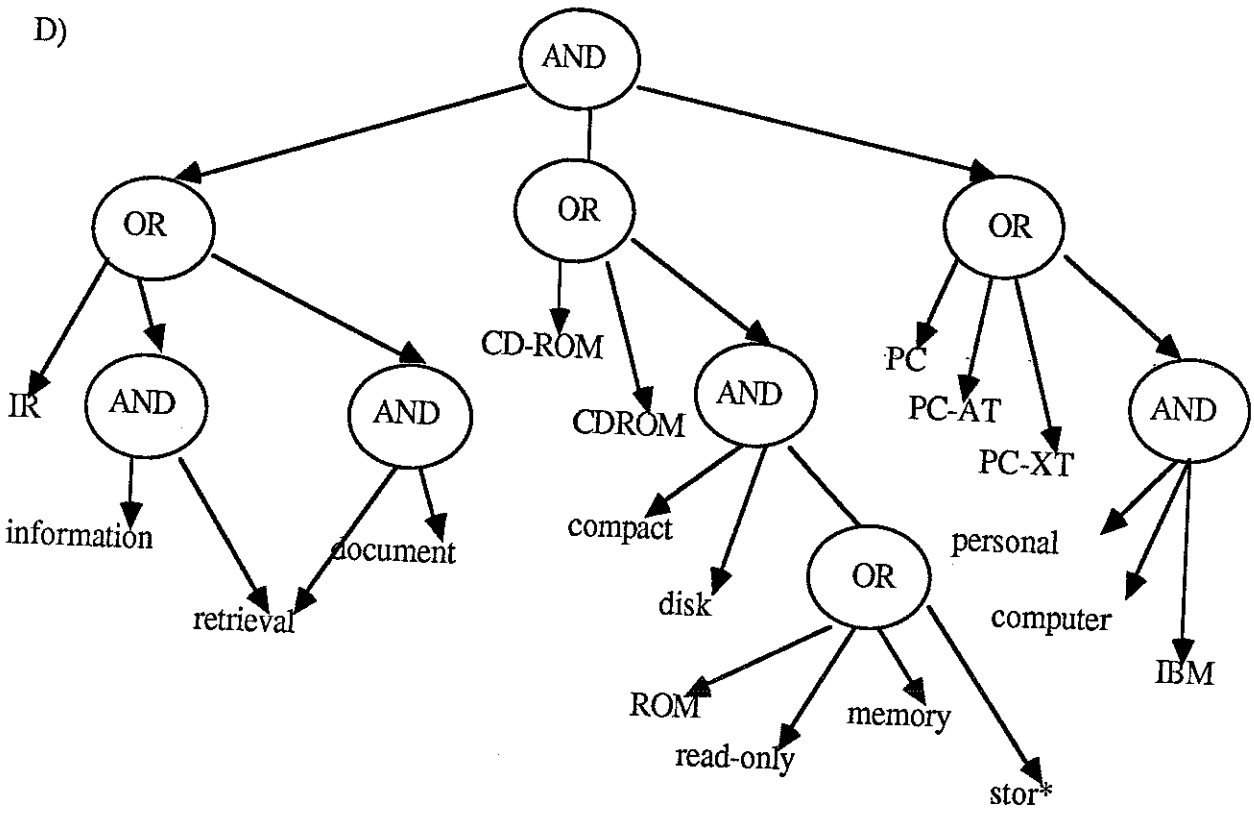
B)

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

D)



It should be noted that the original form of part (a) could not be directly utilized. Indeed, it is difficult to construct good Boolean queries, and so "search intermediaries" are often trained to carry out online searching [57]. They add in terms not included (eg., *document*), provide synonyms ((eg., *PC*) or alternate spellings eg., *CDROM*), truncate words to allow for morphological variants (eg., *stor**), drop function or high-frequency terms (eg., *with, systems*), and add in Boolean operators. It is rarely necessary to use NOT. OR is required when any type of "searchonym" [2] is involved, while AND must be used sparingly so as not to diminish recall. When high precision is needed, metrical or proximity or adjacency operators are also employed, since it is less likely for terms to appear near each other than to simply both occur in the same document.

The entire process of carrying out a Boolean search is summarized in Figure 10. 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. A part of that latter dialog is shown in Figure 11. A retrieval system accepting Boolean queries typically includes an "inverted file" which supplies the "postings" information shown in Figures 10 and 11. Thus, whereas the document file is stored as a set of documents, each of which has words, the inverted file (IF), depicted in Figure 12, turns that organization upside-down. For each term (eg., key word, 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 postings). From Figure 11 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 co-occur. A Boolean query thus will often evolve in steps, and while the final query will always retrieve a reasonable number of "hits," it may have a very complex syntactic and bizarre semantic structure.

FIGURE 10
BOOLEAN QUERY FORMATION AND INVERTED FILE SEARCH

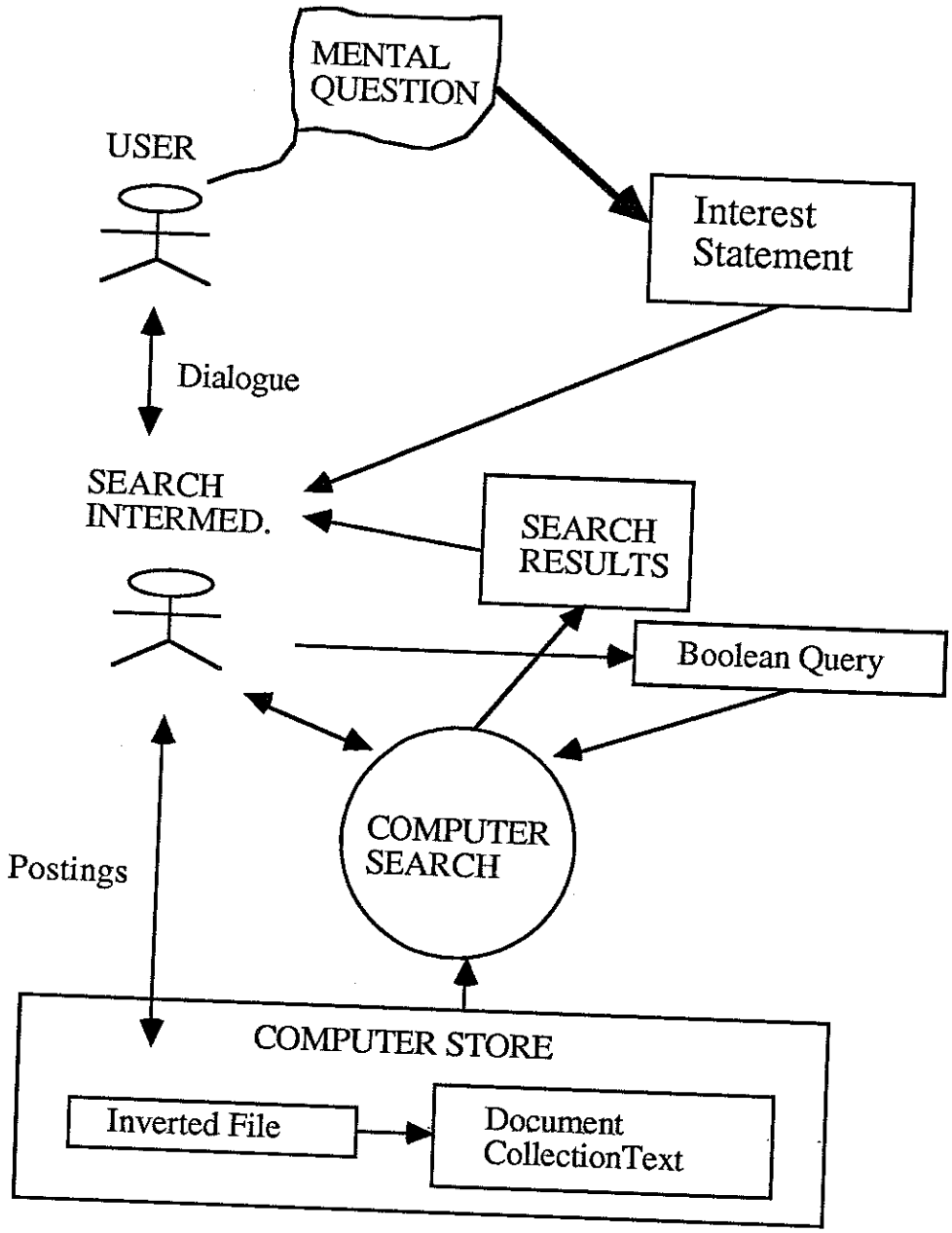
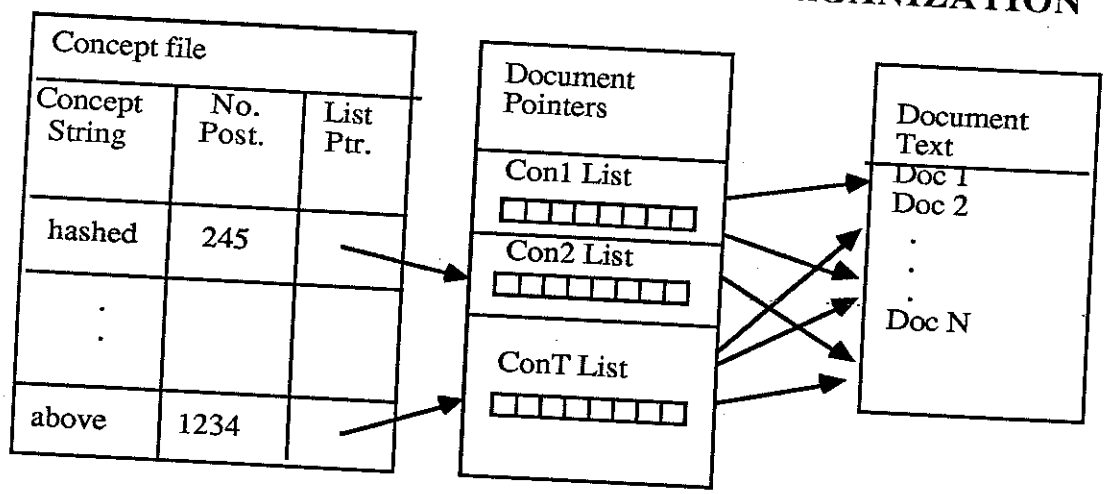


FIGURE 11
INITIAL SECTION OF DIALOG FOR BOOLEAN QUERY PROCESSING

> information
1) INFORMATION 2300
> retrieval
2) RETRIEVAL 749
> document
3) DOCUMENT 892
> 1 AND 2
4) 1 AND 2 539
> 3 AND 2
5) 2 AND 3 372
> IR
6) IR 452
> 4 OR 5
7) 4 OR 5 627
> 7 OR 6
8) 6 OR 7 704

> CD-ROM
9) CD-ROM 421
> CDROM
10) CDROM 121
> compact
11) COMPACT 82
> disk
12) DISK 1583
> 11 AND 12
13) 11 AND 12 48
> ROM
14) ROM 567
> read-only
15) READ-ONLY 391
> memory
16) MEMORY 3865

FIGURE 12. SIMPLE INVERTED FILE ORGANIZATION



In spite of the ubiquitous nature of Boolean query systems for IR, there are many limitations and problems with such systems. As can be seen in Figure 11, 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 in part to explore the effects of ranking the set of documents retrieved by a Boolean query [61]. 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 [55]. A variety of formula for ranking the retrieved documents were compared in order to group similar schemes and identify the best ones [56].

Even in SIRE, however, the searcher cannot specify which terms are most important. One notation for expressing relative importance was proposed in [5]. When Boolean logic is generalized so that there are degrees of truth in the range of zero to one, 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 13 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 close to the top of the list; more documents will typically be retrieved as well, aiding with recall.

FIGURE 13. Sets of Retrieved Documents

A. FUZZY SET

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

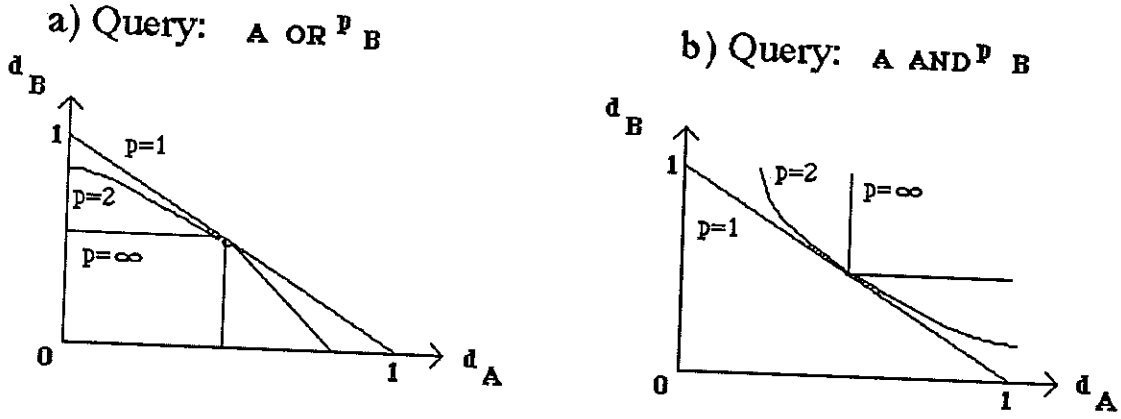
B. BOOLEAN

Document	Similarity
ID003	1.0
⋮	⋮
ID029	1.0
ID003	0.0
⋮	⋮
ID897	0.0

Such a ranking can be produced when Boolean queries are interpreted according to the "p-norm" scheme [80]. 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 [64]). 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. To quantify the degree of strictness, 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 L_p family of norms is employed to measure distance from a suitable ideal point (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 14. Note that p=2 specifies standard Euclidean distance. The curves shown connect points for documents that have equal similarity to the given two-term queries [27]. 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.

FIGURE 14. P-Norm Equi-Similarity Contours



Experiments indicate that the p-norm scheme leads to more effective retrieval than traditional Boolean systems [80]. P-norm queries can also be automatically constructed from simple lists of keywords [79]. When a user is presented with the first ten or twenty documents retrieved by a p-norm or Boolean search, and indicates which of those are relevant, automatic construction of a new "feedback" query is possible too. Typically, this query is better than the original query [81].

3.3 Vector and Probabilistic

The vector and probabilistic models both developed out of early investigations into the statistical characteristics of text collections. Specifically, the vector space model is based on the

observation that the frequencies of occurrence of terms are indicative of their importance [76]. Similarly, the probabilistic model relies on Bayesian theory, presuming 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 [70]. Both of these models are discussed at length in [78]; the description below deals only with the key features.

Figures 15 and 16 illustrate the vector space model, giving examples of spaces for 3 terms and for T terms, respectively. Three of the terms from the query in Figure 9a are chosen in Figure 15, where documents are represented by points whose location is 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. In Figure 16, 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 15
VECTOR SPACE MODEL: 3-D EXAMPLE

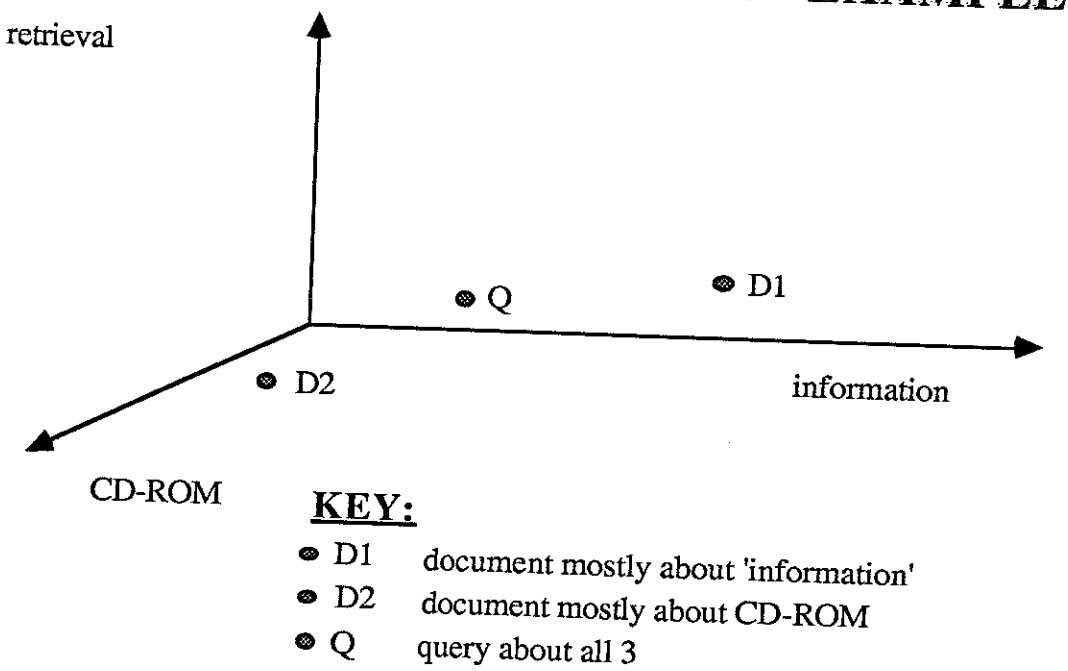


FIGURE 16
VECTOR SPACE MODEL: T-D EXAMPLE

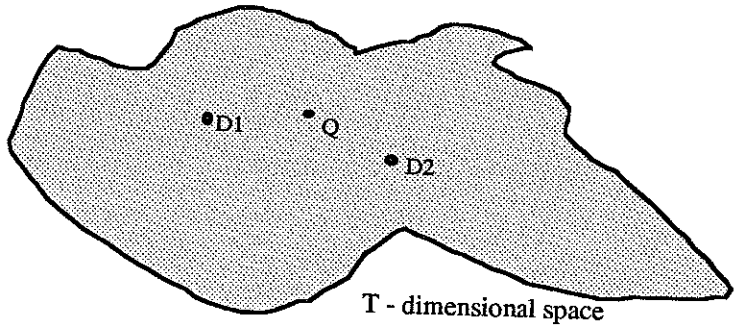


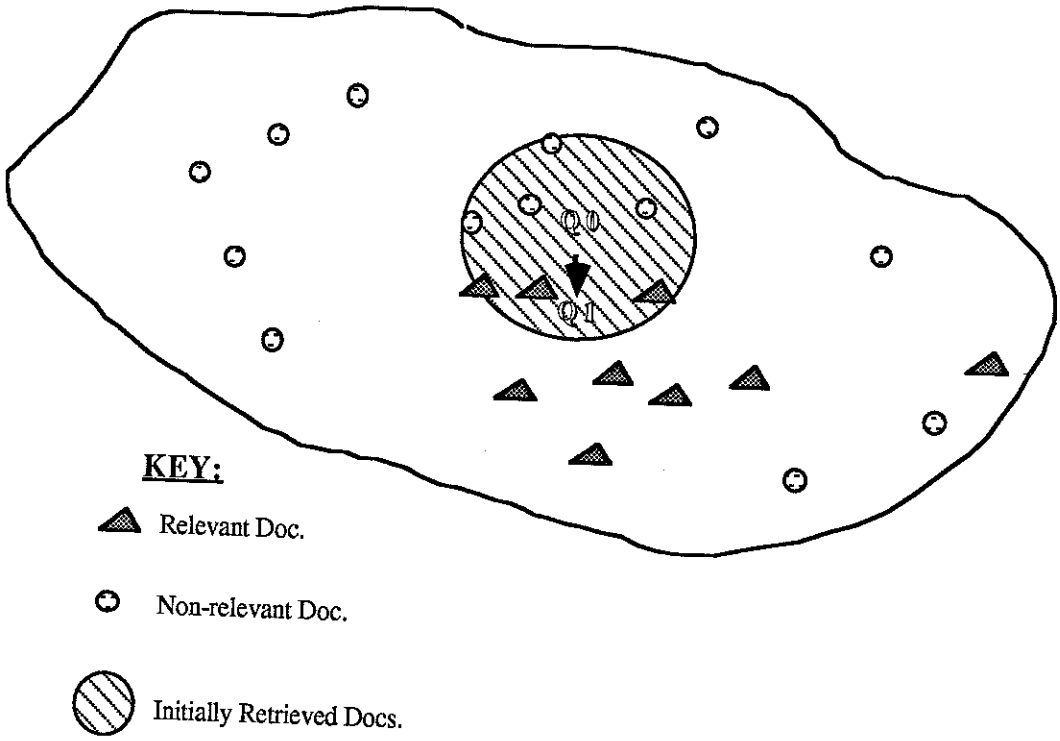
Figure 17 shows one way to represent the vector space, using matrix notation. Each of the T concepts present is associated with a column, and each of the N rows defines a collection document. The value for document I and term J, d_{ij}, indicates the importance of that term's appearance. If the cosine 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/DF_j, where the DF is the number of documents in which the term appears). A similar value can be used for p-norm computations, except that normalization to the range of 0 to 1 is required [27]. In any case, the document-term matrix in practice is very large, but since it is sparse 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 17. DOCUMENT-TERM COLLECTION MATRIX

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

Figure 18 idealizes the vector feedback process [71]. 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.

FIGURE 18. VECTOR FEEDBACK



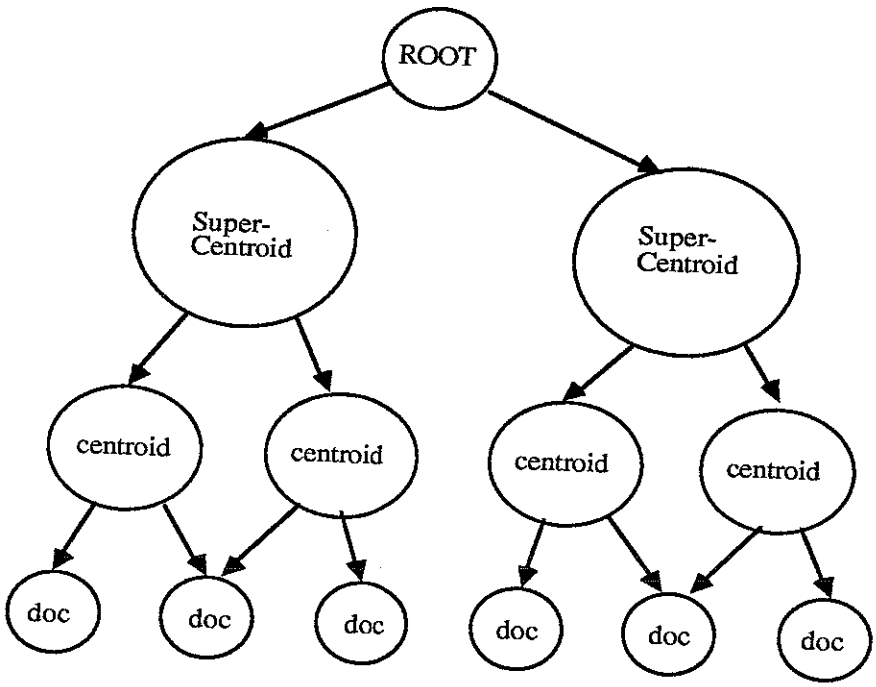
Metrical feedback follows a similar model, but also considers the number of intervening words between occurrences of terms that are to be added to the original query [2]. Probabilistic feedback has like effect, but the construction of Q1 is instead based on a term relevance [70] or term precision formula [75]. Probabilistic weighting schemes are discussed in more detail in [92]. An early implementation of probabilistic retrieval for a practical retrival system is described in [67]. Probabilistic feedback has been incorporated in recent versions of the SMART system [9], and has been adapted for several p-norm algorithms [27]. It is also 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 [59].

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, described in [86], 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 semi-automatic selection of term classes when collections are not too large to warrant the relatively expensive processing.

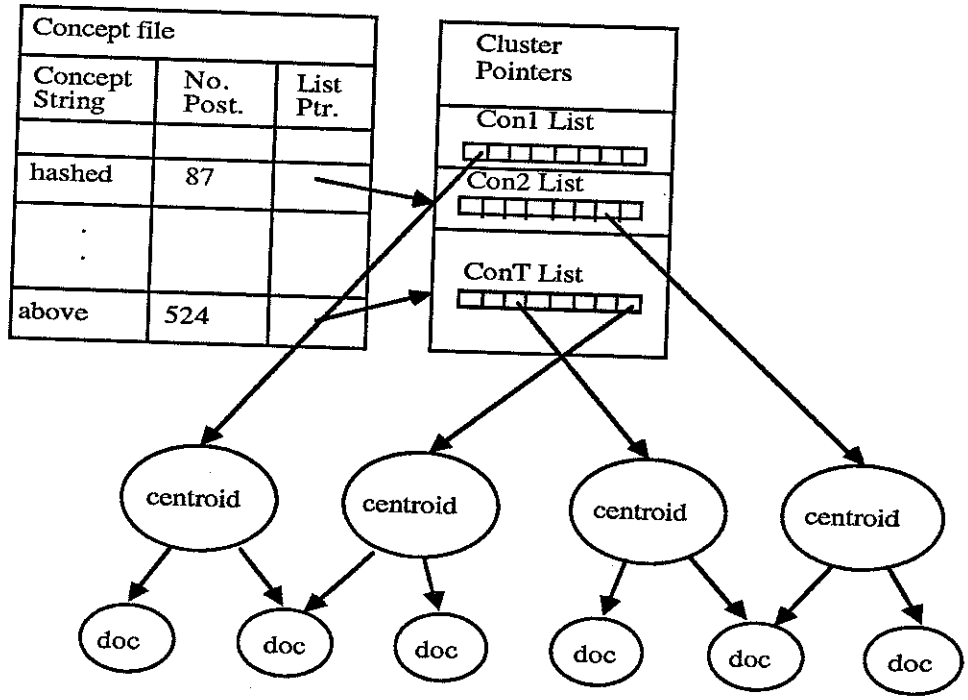
Document clustering has been more thoroughly studied and shows greater promise. Figure 19 illustrates the process, where a number of similar documents are described by a centroid (named since the vector centroid can be so employed), where similar centroids are grouped under super centroids, etc. It is possible for the clustering to be overlapping, as shown for the second and fifth documents, since documents may often be about several topics. 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 19. CLUSTER HIERARCHY ORGANIZATION



Clustering algorithms have long been used in taxonomic analysis [40]. Clustering using the well known single-link method to build searchable hierarchies was discussed in [46]. For large collections, faster methods which still yield hierarchies that can be searched were also developed (eg., [20], [96]). Since cluster centroids are often quite large, top down searching may be very expensive, so the bottom up or hybrid scheme shown in Figure 20 was developed [18]. A recent thorough examination of various clustering approaches 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 [93].

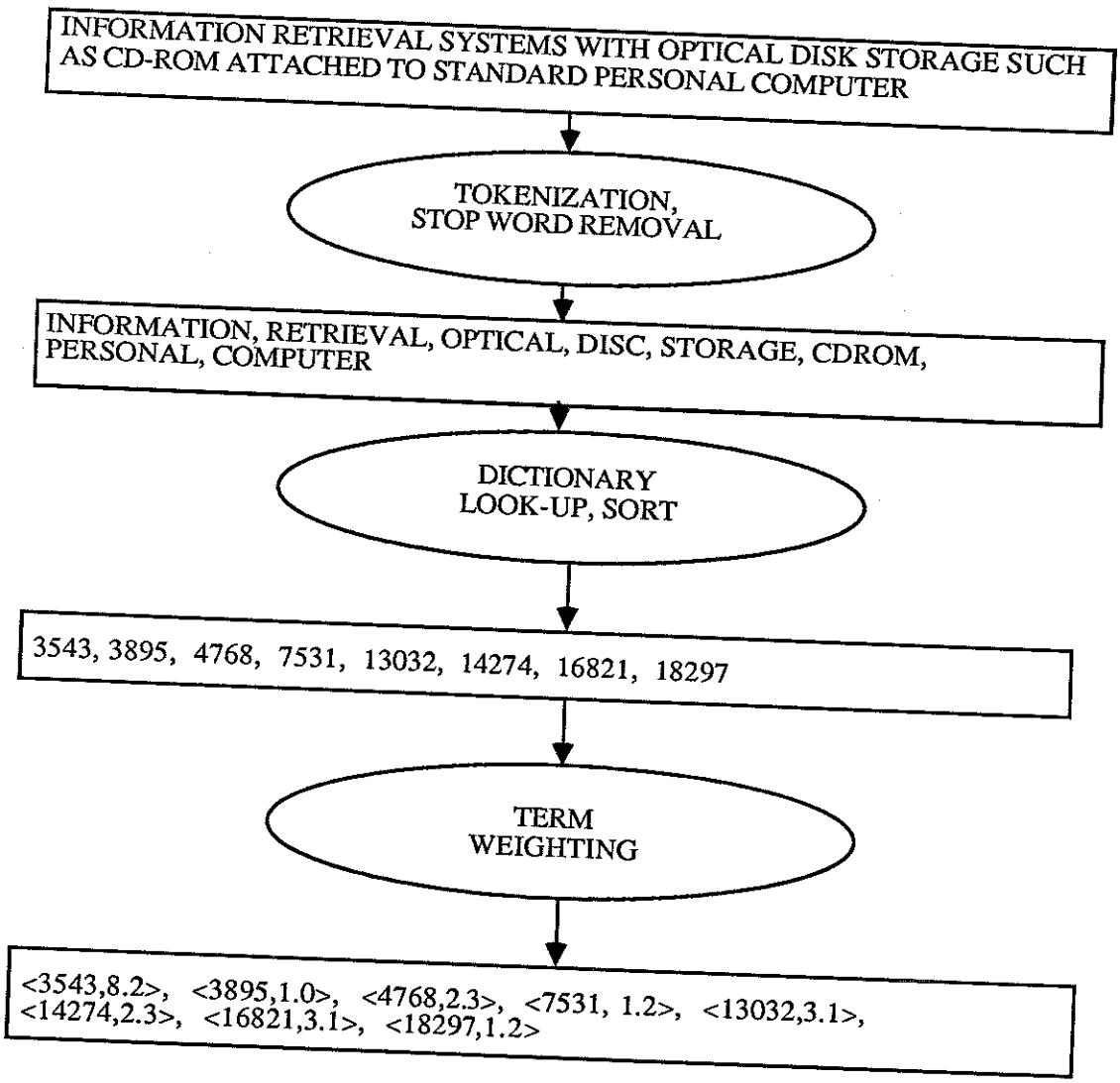
FIGURE 20
HYBRID INVERTED AND CLUSTERED ORGANIZATION



Vector, probabilistic, and p-norm schemes all presume that the document collection has been automatically indexed. Figure 21 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

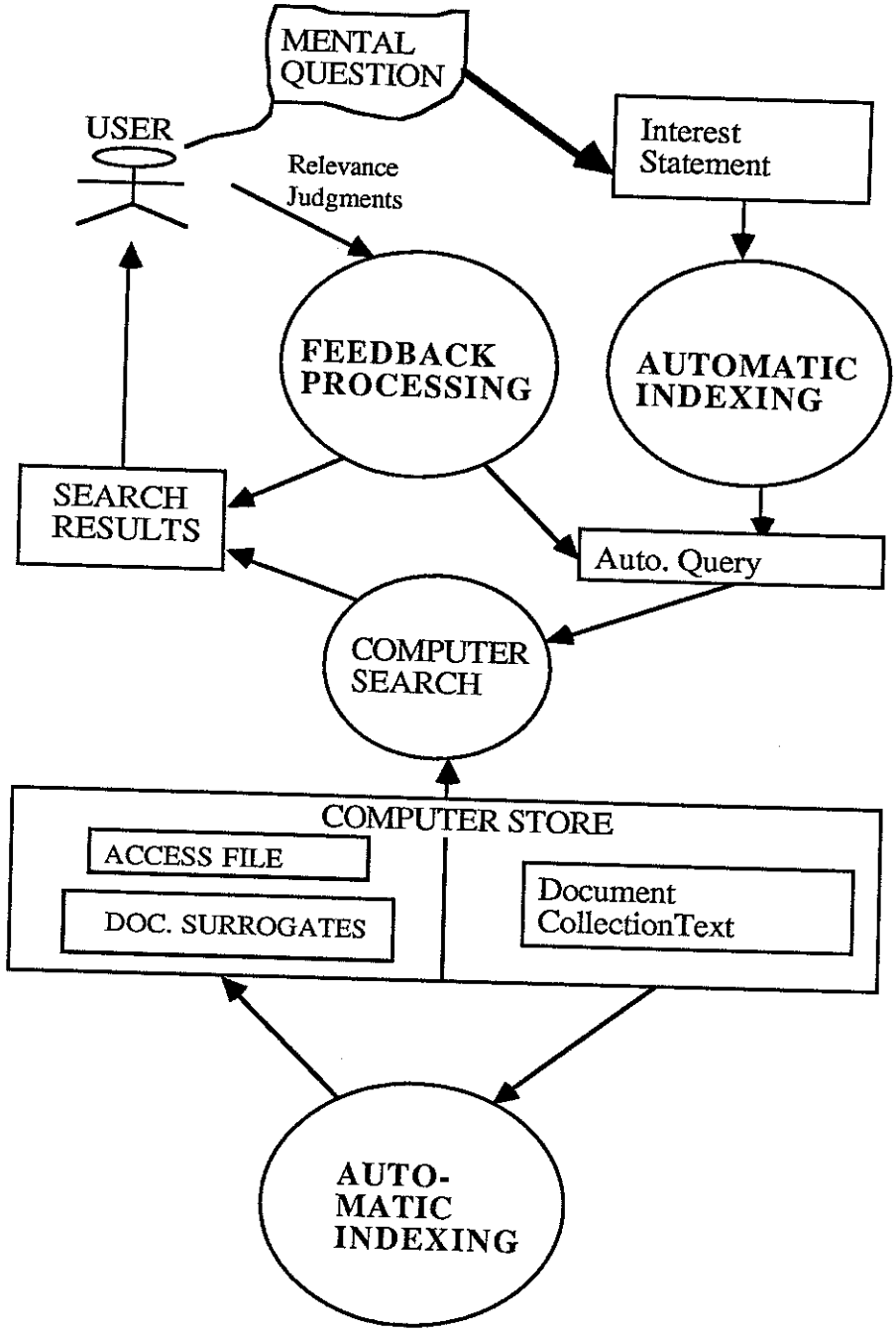
hashing scheme can be employed to identify stop words. The remaining terms are then alphabetized and repetitions are avoided by the inclusion of frequency counts. Normal words are stemmed by a techniques such as that described in [52]. The system dictionary is then searched; new terms in documents are 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 above), 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 21. 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 22.

FIGURE 22
AUTOMATIC INDEXING
AND
FEEDBACK RETRIEVAL



3.4 AI

The field of artificial intelligence has many sub-areas related to IR. Indeed, a number of IR researchers have worked on problems similar to those studied under AI. For example, O'Connor has employed linguistic methods and discourse analysis concepts, along with suitable heuristics, in order to select "answer passages" from texts [62]. Oddy 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 [63]. Hahn and Reimer use word expert parsing routines and discourse models, in order to build a knowledge base that condenses text so it can be more easily browsed and searched [38]. Expert systems have likewise been employed to extend the work of Marcus on automating the job of search intermediaries [97].

At the most fundamental level, AI and IR are both concerned with the analysis and representation of the information content in texts. Thus, Evens' early work on lexicons was in the computational linguistics area of AI [24], but has since shown closer ties to IR. Related work by Fox, for example, has demonstrated the value of relational models of the lexicon to aid in IR thesaurus class utilization [30]. Fox is now involved in extending earlier work on machine readable dictionaries [1] to allow several full dictionaries to be automatically analyzed, so that a comprehensive lexicon can be build 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 recently developed taxonomy of the representation schemes used in IR [85] makes this point clearer, and highlights the need for more IR work in document analysis. The dissertation of Kimura, in the domain of text processing, is of particular interest in providing a framework for modeling large documents [49].

During the last decade, the frame model of Minsky [58] 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 [69].

Similar in form, but more closely tied to the type of cognitive science work that has evolved at Yale, are scripts [82] and more recent constructs for modeling memory. AI methods for handling temporal data have proved useful for certain IR applications [98], 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 [65]. One version of Prolog has been used for computational analysis of stories [16]. Simple but powerful routines for searching small document collections, for building schema representations of texts, and for question answering have also been demonstrated [83].

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 [42]. Much of the progress to date relies on utilizing systems that have been tailored for a particular sublanguage [73]. Less detailed analysis is possible, but requires a knowledge base for whatever subject areas are to be found in the text collection [22]. The problem of recognizing the context for a discussion, so that appropriate specialized knowledge bases can be accessed, is unfortunately very hard to solve [11]. Building schemas to represent story structure is also difficult [72]. Clearly, developing systems that can realistically "comprehend" natural language texts is a difficult research problem [68].

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 [6]. 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 [95], building a UNIX consultant.

The recent emphasis on constructing expert systems [43] is another AI trend that promises to be helpful in retrieval. Several efforts aim to capture the expertise of search intermediaries in

deealing with many different information retrieval services, and in constructing Boolean queries [97]. A more ambitious effort is the integration of browsing with automatic retrieval, where the system has limited models of user and special knowledge about the appropriateness of a given retrieval model and search strategy in each situation [91].

3.5 CODER: Integration of Models

The CODER (COmposite Document Effective/Extended/Expert Retrieval) Project is a comprehensive effort to bring together the best aspects of browsing, Boolean, p-norm, vector, probabilistic, and AI models into a new generation retrieval system [31]. The system is being developed partially in C, for special control and interface portions, but mostly in Prolog. Spcecifically, 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 [60]. The CODER effort includes a number of important components:

1) Constructing a comprehensive lexicon, by automatically analyzing several full English dictionaries.
2) Developing an architecture based on the blackboard model pioneered in the Hearsay-II project [23].
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 posted 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 23.

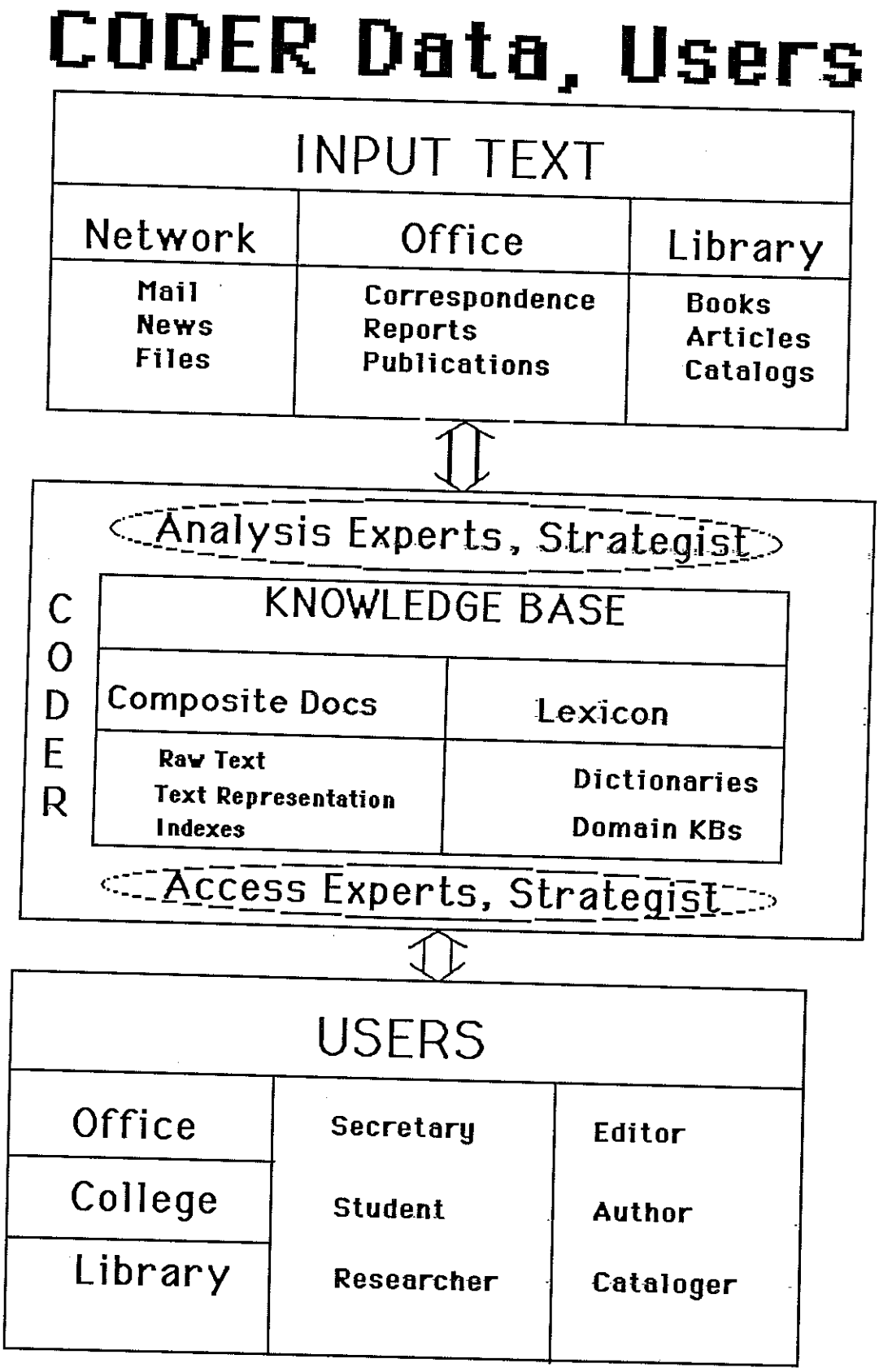

CODER Data, Users
INPUT TEXT
Network
Office
Library
Mail
News
Files
Correspondence
Reports
Publications
Books
Articles
Catalogs
Analysis Experts, Strategist
CODER
KNOWLEDGE BASE
Composite Docs
Lexicon
Raw Text
Text Representation
Indexes
Dictionaries
Domain KBs
Access Experts, Strategist
USERS
Office
College
Library
Secretary
Student
Researcher
Editor
Author
Cataloger

FIGURE 23

4 PAST, PRESENT, FUTURE

4.1 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, or MEDLARS. Information suppliers obtain their data from information providers.

Some of the databases available have been indexed, and search often requires the aid of trained intermediaries to understand specialized thesauri 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, etc. The investment in current retrieval software and emphasis on providing fast and low cost services have made such enhancements irrelevant to managers of most online services.

4.2 CD-ROM Systems Being Developed

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, down-loading 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.

4.3 Future Possibilities

The challenge raised in this report 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 [37], and today, with optical disks such as 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 et al. which relies on studies of the distributions found in existing collections [89]. But it is likely that many research methods can be applied without a great deal of further adaptation.

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 enhanced Boolean query can both 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!

BIBLIOGRAPHY

1) Amsler, R.A. Machine-Readable Dictionaries. *ARIST*, 19, 1984, 161-209.
2) Attar, R. and Aviezri S. Fraenkel. Local Feedback in Full-Text Retrieval Systems. *J. ACM*, 24(3), July 1977, 397-417.
3) Bayer, R. and E. McCreight. Organization and Maintenance of Large Ordered Indexes. *Acta Informatica*, 1(3), 1972, 173-189.
4) Bichteler, J. and Eaton III, E.A. The Combined Use of Bibliographic Coupling and Cocitation for Document Retrieval. *J. Am. Soc. Inf. Sci.*, 31(4), July 1980.
5) Blair, D.C. and M.E. Maron. An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. *Commun. ACM*, 28(3), March 1985, 289-299.
6) Bookstein, A. Fuzzy Requests: An Approach to Weighted Boolean Searches. *J. Am. Soc. Inf. Sci.*, 31(4), July 1980, 240-247.
7) Borgman, Christine L. Psychological Research in Human-Computer Interaction. *ARIST*, 19, 1984, 33-64.
8) Borko, H. and C.L. Bernier. *Indexing Concepts and Methods*. Academic Press, New York, 1978.
9) Boyer, R.S. and J.S. Moore. A Fast String Searching Algorithm. *Comm. ACM*, 20(10), Oct. 1977, 762-772.
10) Buckley, C. Implementation of the SMART Information Retrieval System. TR 85-686, Cornell Univ., Dept. of Comp. Sci., May 1985.
11) Bush, V. As We May Think. *Atlantic Monthly*, 176, July 1945, 101-108.
12) Charniak, E. Context Recognition in Language Comprehension. In *Strategies for Natural Language Processing*, ed. by Wendy G. Lehnert and Martin H. Ringle, Lawrence Erlbaum Assoc., Hillsdale NJ, 1982, 435-454.
13) Choeka, Y. Computerized Full-Text Retrieval Systems and Research in the Humanities: The Responsa Project. *Computers and the Humanities*, 14(3), Nov. 1980, 153-169.
14) Cleverdon, C.W. and E.M. Keen. Factors Determining the Performance of Indexing Systems. Aslib Cranfield Research Project, Vol. 1 and 2, Cranfield, England, 1968.
15) Comer, D. The Ubiquitous B-tree. ACM Comp. Surveys, 11(2), June 1979, 121-137.
16) Cook, P.R. Electronic Encyclopedias. *Byte*, 9(7), July 1984, 151-170.
17) Correira, A. Computing Story Trees. *Amer. J. Comp. Ling.*, 6(3-4), 1980, 135-149.
18) Crawford, R.G. The Relational Model in Information Retrieval. *J. Am. Soc. Inf. Sci.*, 32(1), 1981, 51-64.
19) Croft, W.B. Organizing and Searching Large Files of Document Descriptions. Dissertation. Cambridge Univ., England, 1978.
20) Croft, W.B. and Pezarro, M.T. Text Retrieval Techniques for the Automated Office. In *Office Information Systems*, ed. by N. Naffah, North- Holland, Amsterdam, 1982, 565-576.
21) Dattola, R.T. 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, NJ, 1971.
22) Dattola, R.T. FIRST: Flexible Information Retrieval for Text. *J. Am. Soc. Inf. Sci.*, 30(1), 1979, 9-14.
23) DeJong, G. An Overview of the FRUMP System. In *Strategies for Natural Language Processing*, ed. by Wendy G. Lehnert and Martin H. Ringle, Lawrence Erlbaum Assoc., Hillsdale NJ, 1982, 149-176.
24) Erman, L.D., Hayes-Roth, F., Lesser, V.R., and D.R. Reddy. The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *ACM Comp. Surveys*, 12, 1980, 213-253.
25) Evens, M.W. and R.N. Smith. A Lexicon for a Computer Question-Answering System. *Am. J. Comp. Ling.*, Microfiche 83, 1979.
26) Faloutsos, C. Access Methods for Text. *ACM Comp. Surveys*, 17(1), March 1985, 49-74.

27) Fox, E.A. 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, 1983.
28) Fox, E.A. Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types. Dissertation, Cornell University, University Microfilms Int., Ann Arbor MI, Aug. 1983.
29) Fox, E.A. Some Considerations for Implementing the SMART Information Retrieval System under UNIX. TR 83-560, Cornell Univ., Dept. of Comp. Sci., Sept. 1983.
30) Fox, E.A. Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts. TR 83-561, Cornell Univ., Dept. of Comp. Sci., Sept. 1983.
31) Fox, E.A. Improved Retrieval Using a Relational Thesaurus for Automatic Expansion of Boolean Logic Queries. *Proc. Workshop on Relational Models of the Lexicon*, Stanford Univ., June 29, 1984.
32) Fox, E.A. Composite Document Extended Retrieval: An Overview. In *Res. & Dev. in Inf. Ret.*, Eighth Annual Int. ACM SIGIR Conf., Montreal, June 5-7, 1985, 42-53.
33) Frei, H.P. and Jauslin, J.F. Graphical Presentation of Information and Services: A User Oriented Interface. Inf. Tech.: Res. Dev., 2(1), Jan. 1983, 23-42.
34) Frei, H.P. and Jauslin, J.F. 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, 1984, 383-396.
35) Fujitani, L. Laser Optical Disk: The Coming Revolution in On-Line Storage. *Commun. ACM*, 27(6), June 1984, 546-554.
36) Goldberg, A., Robson, D., Ingalls, D.H.H. *Smalltalk-80: the language and its implementation*. Addison-Wesley, Menlo Park, CA, 1982.
37) Goldberg, A., Robson, D., Ingalls, D.H.H. *Smalltalk-80: the interactive programming environment*. Addison-Wesley, Menlo Park, CA, 1982.
38) Goldstein, C.M. Computer-Based Information Storage Technologies. *ARIST*, 19, 1984, 65-96.
39) Hahn, U. and Reimer, U. 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, 1984, 143-163.
40) Hall, P.A.V. and Dowling, G.R. Approximate String Matching. *ACM Comp. Surveys*, 12(4), 1980, 381-402.
41) Hartigan, J.A. *Clustering Algorithms*. John Wiley and Sons, New York, 1975.
42) Haskin, R. Hardware for Searching Very Large Text Databases. In *Proc. 5th Workshop on Comp. Arch. for Non-Numeric Proc.*, ACM, New York, March 1980, 49-56.
43) Hayes, P. and Mouradian, G.V. Flexible Parsing. *Amer. J. Comp. Ling.*, 7(4), 1981, 232-242.
44) Hayes-Roth, F., Waterman, D.A. and Lenat, D.B., eds. *Building Expert Systems*, Addison-Wesley, Reading, MA, 1983.
45) Heap, H.S. *Information Retrieval, Computational and Theoretical Aspects*. Academic Press, New York., 1978.
46) Hollaar, L.A. A Testbed for Information Retrieval Research: The Utah Retrieval System Architecture. *Proc. 8th Annual. Int. ACM SIGIR Conf. on R&D in Inf. Ret.*, Montreal, June 5-7, 1985, 227-232.
47) Jardine, N. and C.J. Van Rijsbergen. The Use of Hierarchic Clustering in Information Retrieval. *Inf. Stor. and Ret.*, 7(5), Dec. 1971, 217-240.
48) Jennings, M. The Electronic Manuscript Project. *Bulletin Am. Soc. Inf. Sci.*, 10(3), Feb.91984, 11-13.
49) Katzer, J., et. al. A Study of the Overlap Among Document Representations. Syracuse Univ. Sch. of Info. Studies, 1982.

50) Kimura, G.D. A Structure Editor and Model for Abstract Document Objects. Dissertation. Tech. Report No. 84-07-04, Dept. of Comp. Sci., Univ. Washington, July 1984.
51) Lancaster, F.W. Toward Paperless Information Systems. Academic Press, New York, 1978.
52) Lehnert, W.G. The Process of Question Answering: A Computer Simulation of Cognition. Lawrence Erlbaum Assoc., Hillsdale, NJ, 1978.
53) Lovins, B.J. Development of a Stemming Algorithm. *Mech. Trans. and Comp. Ling.*, 11(1-2), March-June 1968, 11-31.
54) Macleod, I.A. and Crawford, R.G. Document Retrieval as a Database Application. Inf. Tech.: Res. Dev. Applications, 2(1), Jan. 1983, 43-60.
55) McGill, M.J., et al. Syracuse Information Retrieval Experiment (SIRE): Design of an On-Line Bibliographic Retrieval System. *ACM SIGIR Forum*, 10(4), Spring 1976, 37-44.
56) McGill, M.J., Koll, M. and Noreault, T. *An Evaluation of Factors Affecting Document Ranking By Information Retrieval Systems*. Syracuse Univ. Sch. of Info. Studies, 1979.
57) Meadow, C.T. and P.A. Cochrane. *Basics of Online Searching*. John Wiley and Sons, New York, 1981.
58) Minsky, M. A Framework for Representing Knowledge. In *The Psychology of Computer Vision*, ed. by P. Winston, McGraw-Hill, New York, 1975.
59) Morrissey, J. 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, 1983, 38-50.
60) Naish, L. *MU-Prolog 3.1db Reference Manual*. Melbourne Univ., 1984.
61) Noreault, T., M. Koll and M.J. McGill. Automatic Ranked Output from Boolean Searches in SIRE. *J. Am. Soc. Inf. Sci.*,28(6), Nov. 1977, 333-339.
62) O'Connor, J. Answer-Passage Retrieval by Text Searching. *J. Am. Soc. Inf. Sci.*, 31(4), 1980, 227-239.
63) Oddy, R.N. Information Retrieval Through Man-Machine Dialogue. *J. Doc.*, 33(1), March 1977, 1-14.
64) Paice, C.D. Soft Evaluation of Boolean Search Queries in Information Retrieval. *Inf. Tech.: Res. Dev. Applications*, 3(1), 1984, 33-42.
65) Pereira, F. Logic for Natural Language Analysis. Tech. Note 275, SRI Int., Jan. 1983.
66) Pfaltz, J.L., Berman, W.H., and E.M. Cagley. Partial-match retrieval using indexed descriptor files. *Commun. ACM*, 23(9), Sept. 1980, 522-528.
67) Porter, M.F. Implementing a Probabilistic Information Retrieval System. *Inf. Tech.: Res. Dev.* , 1(2), 1982.
68) Riesbeck, C.K. Realistic Language Comprehension. In *Strategies for Natural Language Processing*, ed. by Wendy G. Lehnert and Martin H. Ringle, Lawrence Erlbaum Assoc., Hillsdale NJ, 1982, 435-454.
69) Ritchie, G.D. and Hanna, F.K. Semantic Networks - a General Definition and a Survey. *Inf. Tech.: Res. Dev. Applications*, 3(1), 1984, 33-42.
70) Robertson, S.E. and K. Sparck Jones. Relevance Weighting of Search Terms. *J. Am. Soc. Inf. Sci.*, 27(3), 1976, 129-146.
71) Rocchio, Jr., J.J. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System, Experiments in Automatic Document Processing*, ed. by G. Salton, Prentice Hall, Englewood Cliffs, NJ, 1971.
72) Rumelhart, D.E. Notes on a Schema for Stories. In *Representation and Understanding*, ed. by D.G. Bobrow and A. Collins, Academic Press, New York, 1975, 211-236.
73) Sager, N. Sublanguage Grammars in Science Information Processing. *J. Am. Soc. Inf. Sci.*, 26(1), Jan.-Feb. 1975, 10-16.
74) Salton, G. A New Comparison Between Conventional Indexing (Medlars) and Text Processing (SMART). *J. Am. Soc. Inf. Sci.*, 23(2), 1972, 75-84.
75) Salton, G., Yang, C.S., and C.T. Yu. A Theory of Term Importance in Automatic Text Analysis. *J. Am. Soc. Inf. Sci.*, 26(1), Jan.-Feb.1975, 33-44.

76) Salton, G., Wong, A., and C.S. Yang. A Vector Space Model for Automatic Indexing, *Commun. ACM*, 18(11), Nov. 1975, 613-620.
77) Salton, G. The SMART System 1961-1976: Experiments in Dynamic Document Processing. In *Encyclopedia of Library and Information Science*, 1980, 1-36.
78) Salton, G. and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Newe York, 1983.
79) Salton, G., Buckley, C., and E.A. Fox. Automatic Query Formulations in Information Retrieval. *J. Am. Soc. Inf. Sci.*, 34(4), July 1983, 262-280.
80) Salton, G., Fox, E.A., and Wu. H. Extended Boolean Information Retrieval. Commun. ACM, 26(11), Nov. 1983, 1022-1036.
81) Salton, G., Fox, E.A. and E. Voorhees. Advanced Feedback Methods in Information Retrieval, *J. Am. Soc. Inf. Sci.*, 36(3), 1985, 200-210.
82) Schank, R.C. and R.P. Abelson. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Assoc., Hillsdale, NJ, 1977.
83) Simmons, R.F. *Computations from the English.* Prentice-Hall, Englewood Cliffs, NJ, 1984.
84) Small, H. Co-Citation Context Analysis and the Structure of Paradigms. *J. Doc.*, 36(3), Sept. 1980, 183-196.
85) Smith, L.C. and Warner, A. J. 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, 1984, 31-49.
86) Sparck Jones, K. *Automatic Keyword Classifications.* Butterworths, London, 1971.
87) Sparck Jones, K. and C.J. Van Rijsbergen. Information Retrieval Test Collections. *J. Doc.*, 30(4), March 1976.
88) Stonebraker, M., et al. Document Processing in a Relational Database System. *ACM Trans. on Office Inf. Systems*, 1(2), April 1983.
89) Tague, J., Nelson, M., and H. Wu. Problems in the Simulation of Bibliographic Retrieval Systems. In *Information Retrieval Research*, ed. R.NM. Oddy, S.E. Robertson, C.J. Van Rijsbergen, and P.W. Williams, Butterworths, London, 1981, 236-255.
90) Tenopir, Carol. Full-Text Databases.*ARIST*, 19, 1984, 215-246.
91) Thompson, R.H. and W.B. Croft. An Expert System for Document Retrieval. *Proc. Expert Systems in Gov. Symp.*, IEEE, Oct. 1985, 448-456.
92) Van Rijsbergen, C.J. *Information Retrieval: Second Edition.* Butterworths, London, 1979.
93) Voorhees, E.M. The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval. Dissertation. TR 85-705, Cornell Univ., Dept. of Comp. Sci., Oct. 1985.
94) Weyer, S.A. Searching for Information in a Dynamic Book. SCG-82-1. Xerox PARC, Palo Alto, CA, Feb. 1982.
95) Wilensky, R., Arens, Y., and D. Chin. Talking to UNIX in English: An Overview of UC. *Commun. ACM*, 27(6), 1984, 574-593.
96) Williamson, R.E. Real-Time Document Retrieval. Dissertation. Cornell Univ., June 1974.
97) Yip, Man-Kam. An Expert System for Document Retrieval. M.S. Thesis. M.I.T., Cambridge, MA, 1979.
98) Zarri, G.P. An Outline of the Representation and Use of Temporal Data in the RESEDA System. *Inf. Tech.: Res. Dev. Applications*, 2(2/3), July 1983, 89-108.