Chapter 2

Systems Based on Inverted Files

0 PREVIEW

This chapter covers the operations of retrieval systems based on inverted files. The effect of the main Boolean operators that are used in many information retrieval environments is first described, followed by an examination of adjacency and term frequency operations. Selected features from the available commercial information retrieval systems are then described. Finally enhancements are considered to the basic inverted file design.

1 GENERAL CONSIDERATIONS

It was indicated in Chapter 1 that virtually all the commercially available systems are based on inverted file designs [1]. That is, each system logically consists of a document file and one or more auxiliary directories known as the "inverted index." The inverted index contains the allowable indexing terms. For each term an associated list of document reference numbers is included in the index [2]. Each document reference number uniquely specifies a document to which a given term has been assigned. Thus, the retrieval of the documents identified by an arbitrary term requires a search of the index to find the desired term, and hence the associated document reference numbers. Finally the identified documents are selected from the document reference file. This simple process carries out the functions of the SIMILAR operation introduced in Chapter 1 when single-term queries are processed. In practice, query statements consisting of a single term are rarely used. Occasionally a single term may be sufficiently new or unique to make it a reasonable query all by itself [3]. An example of a term which may fit this criterion is "videodisc." However, by the time this text is published, many thousands of references may already exist even for such a specialized term.

A Boolean Expressions

Normally the searcher has to restrict the number of items retrieved by a search to fewer documents than are found by using a single term. For instance, the terms "information" and "retrieval" may be separately included in the inverted index. The person searching for references to the concept "information retrieval"would like to ensure that the retrieved documents are at least identified by both terms. The assumption is that documents that include both terms may in fact be about the topic "information retrieval" [4]. The SIMILAR function of Chapter 1 is then interpreted as requiring the presence of both query terms to select a given document as potentially useful.

To identify those documents containing both the term "information"and the term "retrieval,"it is necessary to process only the information from the inverted index rather than the information from the document file. Boolean logic is used to construct queries consisting of a variety of terms using the Boolean operators AND, OR, and NOT [5]. These operations are implemented by using set intersection, set union, and set difference procedures, respectively. One query which may be used to identify documents on "information retrieval"may be stated as

INFORMATION AND RETRIEVAL

The following procedure might then be used to find the corresponding documents:

1 Use the inverted index to retrieve the document reference numbers associated with the term INFORMATION. Call these document reference numbers Set 1.

2 Use the inverted index to retrieve the document reference numbers associated with the term RETRIEVAL. Call these document reference numbers Set 2.

3 Determine which document reference numbers constitute the intersection of Sets 1 and 2, that is, are contained in both Set 1 and Set 2. Call these document reference numbers Set 3.

4 Use the main document file to retrieve the documents identified by the document reference numbers in Set 3.

In other words, documents included as members in both Set 1 and Set 2 (the set intersection) are those that satisfy the query and are to be retrieved.

The query

INFORMATION OR RETRIEVAL

refers to documents which are identified either by the term INFORMATION or by the term RETRIEVAL or by both terms. Set 1 and Set 2 may be determined in the same manner as for the AND operator. These sets are then combined into a new Set 3 which contains an identifier for each document contained in either Set 1 or Set 2 or in both sets (set union). Any document included in this composite set is then retrieved by the query statement.

The NOT operator is usually implemented as an operator which specifies that some particular term is to appear in the retrieved document but that some other term is not allowed to appear. For instance, the query statement

INFORMATION NOT RETRIEVAL

refers to documents containing the term INFORMATION but not containing the term RETRIEVAL. In order to accomplish the negation operation using an inverted file system, the following procedure may be used:

1 Use the inverted index to retrieve the document reference numbers associated with the term INFORMATION. Call these document reference numbers Set 1.

2 Use the inverted index to retrieve the document reference numbers associated with the term RETRIEVAL. Call these document reference numbers Set 2.

3 Remove from Set 1 any document reference number included in Set 2. That is, construct the set difference between Sets 1 and 2.

4 Use the document file to retrieve the documents indicated by the document reference numbers remaining in Set 1.

In other words, the documents that satisfy the query statement have reference numbers contained in Set 1 but not in Set 2 (set difference).

B Order of Operations

The complexity of a query can grow substantially as new operators are added, and a variety of rules are necessary to ensure that queries submitted by the searcher are interpreted correctly by the retrieval system. Consider as an example the inverted index as shown in Table 2-1. A query such as

APPLE AND ORANGE OR BANANA

is ambiguous. If one starts at the left of the query statement and works toward

Table 2-1 Sample Inverted Index

Terms	Document reference numbers
APPLE	1 3 5 7
ORANGE	2 3 4 5 6
BANANA	4 6 8
GRAPE	3 7 9 11

the right, the items to be retrieved are identified by the document reference numbers

3 4 5 6 8

because the set intersection between the sets for APPLE and for ORANGE produces items 3 and 5 to which are added items 4, 6 and 8 when the set union is carried out with the BANANA set. On the other hand, if one starts at the right and works toward the left of the query statement, then the items to be retrieved are given by the document reference numbers

3 5

because the union between ORANGE and BANANA produces items 2, 3, 4, 5, 6, and 8 and the intersection that follows with the set for APPLE restricts the output to 3 and 5.

The order in which the operations are carried out is critical. The strategy may be left-to-right or right-to-left, or some other method may be used to specify the order in which the operations are to be executed. For instance, one procedure specifies that all the OR operators are performed first, followed by the AND operators, and finally the NOT operators; all equivalent operators are performed from left to right. Parentheses are usually provided to circumvent the strict processing order described above. In particular, operations within parentheses are normally completed first. For the previously used example the left-to-right order is thus equivalent to (APPLE AND ORANGE) OR BANANA, whereas the right-to-left order corresponds to APPLE AND (ORANGE OR BANANA). Each operation or set of operations within parentheses is first carried out according to the regular processing rules. When this is completed, the remainder of the query statement is processed. Consider the query statement:

(APPLE AND ORANGE) OR (BANANA AND ORANGE)

The following process may be used for this statement:

1 The first AND operator on the left side of the query statement combines the document reference numbers associated with APPLE and ORANGE (items 3 and 5).

2 The second AND operator combines the document reference numbers associated with BANANA and ORANGE (items 4 and 6).

3 Finally the OR operator combines the sets retrieved in steps 1 and 2 (items 3, 4, 5, and 6).

The set of documents retrieved by using the parentheses differs from the sets retrieved by rules given earlier. Once the rule for parentheses has been established, it can be applied over and over again. That is, nested parentheses can be used so that the operations within the innermost pair of parentheses will be carried out first. For instance the query statement

(APPLE AND (ORANGE OR BANANA)) NOT GRAPE

is executed beginning with the (ORANGE OR BANANA) portion of the query statement. In order to allow parentheses, it is necessary to keep track of intermediate results. For this reason, some systems do not allow the use of parentheses and others allow only limited nesting of parentheses.

2 ADJACENCY AND TERM FREQUENCY FEATURES

Each commercial system includes certain features that make it unique. This complicates the problem of learning how to use each of the systems. A few of the more interesting and basic operations are described in the remainder of this chapter. Since the various processing approaches are considered to be proprietary information by the system vendors, there is no assurance that the methods presented here are strictly factual. The presentation of the basic methodologies covered in the next few paragraphs is, however, expected to be reasonably accurate.

A Adjacency Operations

Consider a retrieval system which allows the searcher to formulate queries using words included in the document texts. It may be useful to specify that two words must appear next to each other in a text and in the proper word order. If the operator ADJ stands for adjacency, a query for documents on "information retrieval" may now be stated as

INFORMATION ADJ RETRIEVAL

The searcher is then assured that the two search terms do not appear in unrelated portions of the document but are in fact contained in adjacent word positions.Thus, the probability that the concept "information retrieval" is contained in the document is higher than if the searcher were to use the terms INFORMATION and RETRIEVAL combined by an AND operator.

It is difficult to implement the adjacency operation using the basic definition of the inverted file. The following procedure may be used, however, when only a basic inverted index is available:

1 Use the standard inverted file to identify the documents that satisfy the query

INFORMATION AND RETRIEVAL

2 Use the document file to search specific fields of the corresponding documents by means of a character by character match (a string search) to detect the presence of the characters "INFORMATION RETRIEVAL." The fields to be searched are normally prespecified and may include the title and abstract for each document.

3 Retrieve from the document file those items for which at least one complete match of the given character string is found.

String searching is a laborious task which consists in scanning an arbitrary set of symbols in search of a specific sequence of symbols. That is, the text of a document is examined character by character until the desired sequence of characters is found. String searching procedures are implemented in certain systems but are used only when essential. In particular, the system normally warns the user of the inefficiency of the string searching process, and various restrictions may limit the conditions under which string searching may be conducted.

Another possibility for implementing the ADJ operator consists in enhancing the inverted file by adding information about the location of words within each document. For instance, if the stored documents consist of abstracts of two or more paragraphs, and each paragraph includes several sentences, then the term location information might include the document reference number, paragraph number, sentence number, and word number within each sentence. Thus, RETRIEVAL (345 1 2 5) would indicate that the term RETRIEVAL occurs in the first paragraph, second sentence, and fifth word of document 345. Document 345 would be retrieved by the query statement

INFORMATION ADJ RETRIEVAL

whenever the entry INFORMATION (345 1 2 4) appears in the inverted file in addition to the entry RETRIEVAL (345 1 2 5).

Another way to provide term location information is to add to each term entry in the inverted file a distance indicator specifying for each word occurrence the distance from the beginning of the text in terms of the number of intervening words. Thus, RETRIEVAL (345 13) indicates that the term RETRIEVAL occurs 13 words from the beginning of document 345. Again document 345 would be retrieved by the previous query only if INFORMATION (345 12) were also included in the enhanced inverted file. This procedure does not recognize sentence boundaries and is not therefore completely equivalent to the previous method based on paragraph, sentence and word numbers. In practice, the two methods are probably equivalent in terms of retrieval performance.

A comparison of the methods for carrying out the ADJ operation shows that the character by character analysis of documents requires substantial computational resources but the inverted file is not encumbered with additional word location information. The other two methods are based on an expanded inverted file. When word location information is kept in the inverted file, little extra processing is required for the ADJ operation with the exception of an added comparison between two groups of numbers to determine the appropriate order. The added word location information does, however, take up a great deal of potentially valuable storage space.

To date the tradeoff has seemed clear. The character by character searching, as usually implemented, is so inefficient that the use of extra storage in the inverted file appears mandatory. However, fast string searching operations have recently been discovered that may reverse this situation. These methods will be discussed in Chapter 8 of this volume.

B Frequency Information

Another way of enhancing an inverted file system is to include information about the frequency of occurrence of the individual terms. It has been shown that special usage patterns exist for the words included in natural language texts in certain subject specialties. In particular, the frequency of use of a given term may correlate with some indication of the importance of that term in the given subject area. If word frequency information is to be used in retrieval, it must be stored by the system, and the most practical way to do this is to include the information in the inverted file. In many systems "posting"information is kept in the inverted file for each term. That is, the inverted file includes information about the number of documents in which a given term occurs. This frequency information is referred to as the number of postings for the term. In this way a user can quickly ascertain the number of documents that will be retrieved by using a given term. For instance, the term INFORMATION has 53,504 postings in the ERIC data base available through the DIALOG system at the time this is being written. A query which includes only this single term would therefore retrieve 53,504 documents.

3 COMMERCIAL INVERTED FILE SYSTEMS

A The DIALOG System

The DIALOG system is a product of Lockheed Information Systems of Palo Alto, California. In May of 1980 some 122 individual data bases were available through the DIALOG system [6,7].

The DIALOG system is based on an inverted file design. The system creates sets of document reference numbers by means of a SELECT command.

Thus, the SELECT INFORMATION command creates a set of document reference numbers associated with the term INFORMATION. The system provides the user with a set number identifying these document reference numbers. For example, the statement given earlier would be assigned set number 1 if it were the first SELECT command issued by the searcher. A second command SELECT RETRIEVAL would therefore create set number 2. These sets may then be processed by a COMBINE statement which allows the use of the Boolean operators AND (*), OR (+), or NOT (-). Thus the statements

COMBINE 1 AND 2

or

COMBINE 1 * 2

are used to form the query statement

INFORMATION AND RETRIEVAL

This query was introduced earlier in this chapter when the document reference numbers associated with INFORMATION (Set 1) and the reference numbers associated with RETRIEVAL (Set 2) were combined with an AND operator to form the new Set 3. The NOT operations are performed first in the DIALOG system, followed by the AND operations, and finally by the OR operations. Parentheses are allowed in order to alter this specified sequence of operations. Thus, a statement such as

COMBINE (4 OR 5 OR 6) AND (7 OR 8) NOT 9

produces a legitimate search operation assuming that Sets 4, 5, 6, 7, 8, and 9 have all been previously defined.

In the DIALOG system a term may be truncated on the right to indicate that any characters following the truncation symbol are acceptable. For example, PSYCH? can be used as a search term to retrieve items associated with

PSYCHIATRIST
PSYCHIATRY
PSYCHOLOGICAL
PSYCHOLOGIST
PSYCHOLOGY

and any other terms that begin with the characters PSYCH. The truncation symbol "?" may also be used to specify the maximum number of characters that may appear following the user supplied characters. The number of characters allowed is indicated by the number of "?" symbols immediately after the word followed by a blank and another "?" symbol. For example,

DOCUMENT?? ?

specifies that the search term DOCUMENT may be followed by up to 2 additional arbitrary characters. Thus, documents associated with the terms DOCUMENTS and DOCUMENTED will be retrieved, but documents associated with the term DOCUMENTATION will be rejected.

The truncation operations can be carried out in an obvious manner using an inverted file. In the first case (truncation without limits), the inverted file need only be searched for the terms whose initial characters correspond to the user input. Using the PSYCH? example, the system need only examine the first five characters of the terms included in the index file. If an exact match is found, the associated document reference numbers are placed in the retrieval set. The same logic is used for truncations with a limited number of trailing characters. That is, an exact match of characters is required between the characters supplied by the searcher and the characters of the term stored in the index file. Once this condition is satisfied, it is necessary to consult the index file to determine if the number of additional characters of the given term in the index file meets the criterion specified by the user. In order to determine that the matching term carries the right number of trailing characters, the characters specified by the ? symbols need not be examined individually. However, characters occurring to the right of those specified by the ? symbols must be considered. If no characters occur in the term stored in the index file beyond the last ? symbol, the corresponding document reference numbers are retrieved.

The truncation character "?" may also be embedded inside a term supplied by the user. For example, WOM?N would be used to indicate both the term WOMAN and the term WOMEN. The process used to handle requests of this kind may be based on methods similar to those described earlier. Note that there is no need to add any location or frequency information to the inverted file.

DIALOG also offers the ability to search for pairs of adjacent words or for terms occurring within a specified number of words of another term. This capability is based on the use of terms derived from the actual texts of documents or document abstracts, as opposed to terms assigned from a controlled vocabulary. Controlled terms are not generally assigned in any meaningful sequence. The pertinent operator used is "(W)," and it must be used with the SELECT operator. Thus,

SELECT PROGRAMMING (W) LANGUAGE

would retrieve the documents from a data base which included the terms PROGRAMMING and LANGUAGE occurring side by side in the text and in that stated word order. As in the case of the ADJ operator discussed earlier, the corresponding search can be carried out either by processing the actual texts of documents following identification of items which include both terms, or by adding location information to the inverted file. In the DIALOG system, the inverted file is enhanced by the position number of each word within each document. Thus retrieval is based on the determination of consecutive word position numbers in the inverted file.

When word position information is available, one can also determine if two particular terms occur within a specified number of words of each other in a text. This is done by subtracting the location number of the first term from that of the second one. If the order of occurrence of the terms is deemed important, one may want to insist on a positive difference between location numbers. If the difference is allowed to be either positive or negative, the order of the terms is disregarded. Thus,

```
SELECT PROGRAMMING  (5 W) LANGUAGE
```

would find all documents in which the term LANGUAGE follows the term PROGRAMMING within a distance of up to five words; term order is clearly taken into account. If term order is not important, the term LANGUAGE may either precede the term PROGRAMMING by up to five words, or it may follow PROGRAMMING by up to five words. In the DIALOG system the order of the words is important and the system assumes they are to appear in the order specified in the query statement.

The DIALOG system also uses field identification for author (AU), classification code (CC), corporate source (CS), document type (DT), journal name (JN), language (LA), publication year (PY), and update (UD). The latter field indicates when the document was added to the data base. Since some of these fields contain measurable values, it is possible to include the corresponding values in a search statement. For instance, a range of values can be specified for the publication year as follows:

```
SELECT PY = 1977 : PY = 1979
```

which indicates that documents with a publication year between 1977 and 1979 are acceptable. The colon (:) designates a range of measurable values to be used.

From a computational point of view such a process seems to present problems. If the publication year is included in the inverted file for each document, then in order to select the appropriate range of publication years one must search for exact matches corresponding to the whole range of allowable publication years or be able arithmetically to compare the publication year. Thus, for the example given earlier the specified dates must include 1977, 1978, and 1979. The number of possible representations for a publication date in a given data base is quite large. For example, 1978, June 1978, 6/78, and '78 may all be used to represent publication dates for the year 1978. If all those alternatives were allowed in a retrieval system, a complicated search would become necessary. In the DIALOG system a specific representation is selected, in this example 1978. This makes it possible to carry out exact matches on specific dates to retrieve documents with date specifications. Alternatively, it is quite feasible to keep a separate inverted index for publication dates alone. Assuming that such a file is kept in chronological order and in a numeric representation, the system need only search for values in the range 1977:1979. Thus, a document may have terms or values assigned from a number of different inverted indexes. A different index may be used for each field of the document such as publication year or author. Different indexes may also help to distinguish identical values for different keys such as publication year 1980 and page number 1980.

Many other features are included in the DIALOG system. In this discussion some of the important features have been highlighted to indicate approaches that have been used by the system designers.

*B The STAIRS System

Another prominent system is the storage and information retrieval system (STAIRS), which is a program product of the IBM Corporation. Whereas the STAIRS system itself is available through IBM, no data bases are made available by IBM. Rather the user must purchase or lease the STAIRS programs and apply the system either to commercially available data bases or to private data bases. STAIRS runs on the customer's own computer.

STAIRS consists of two sets of programs:

1 Utility programs for data base creation and maintenance

2 An on-line retrieval system called AQUARIUS, which stands for a query and retrieval interactive utility system.

The retrieval function is a multiuser system which develops an effective dialogue with the user. This dialogue leads to the search and eventual retrieval of stored data.

A principal difference between STAIRS and DIALOG is that a full STAIRS implementation includes not only a text processing and document retrieval function but also an associated data base management system. The latter is designed to process formatted (highly structured) information such as numeric data available in tabular format. In the data management context, the retrieval of records is based on the values of particular attributes of the records. STAIRS uses separate modes of operation to handle text and structured data known as the SEARCH and SELECT modes, respectively.

To use the text retrieval system it is necessary to create an inverted file, a text index, and one or more text files. These are illustrated in Fig. 2-1. The text file contains the documents using a special format in which the retrieved documents are presented to the retrieval system users. The text index includes pointers to records in the text file, as well as privacy information and formatted data associated with subsets of records in the text file. These formatted data

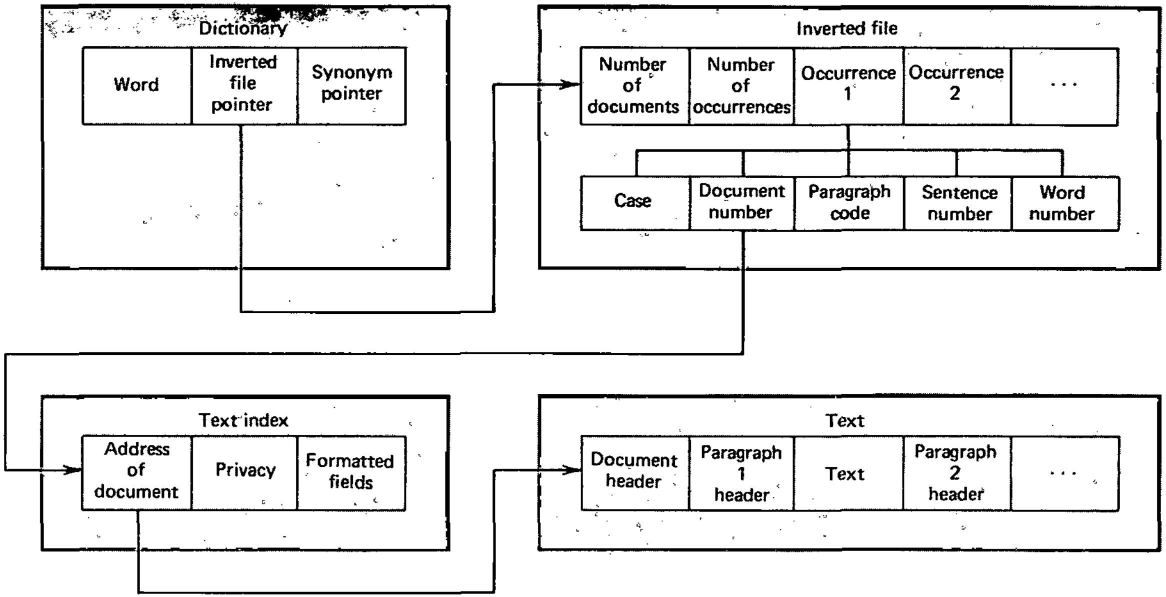

Figure 2-1 STAIRS file organization. (*Adapted from reference 8.*)

include attributes for which a range of values can be specified in advance. Because the allowable values are known in advance, the manner in which the values appear in the files can be prespecified. For instance, the publication dates may be restricted to the format "1978" and dates of the form 6/78 or any other format will be disallowed.

The dictionary contains a record for every unique word included in the data base. Associated with this entry is a pointer to a list identifying each occurrence of the particular word in the text. This is of course the inverted index described earlier in this chapter. The inverted file portion of the system may also be used to identify words which are synonymous with a given term. The system accomplishes this by maintaining a separate synonym dictionary. Access to an individual word in the dictionary is obtained by using letter pairs. That is, the first two letters of any word are used to identify a specific grouping of terms which is searched in order to find the wanted term. Associated with the term is a pointer to the list of associated document reference numbers. The dictionary has two levels: the first level contains the information about the letter pairs and indicates where the search must start on the second level to find the words beginning with particular letter pairs; the second level contains the actual words along with associated word length information and synonym information. Thus the dictionary itself is organized as an indexed file (see Fig. 2-2). Note that the location information in the inverted file consists of three numbers (triplets) representing paragraph code, sentence number, and word number. This information is present for each occurrence of each word in the data base.

The STAIRS retrieval system uses the free text of the documents or document abstracts for search purposes. The location information for each term is an important part of the STAIRS system, as is much of the remaining information described in Fig. 2-1. Controlled vocabulary terms may also be used with

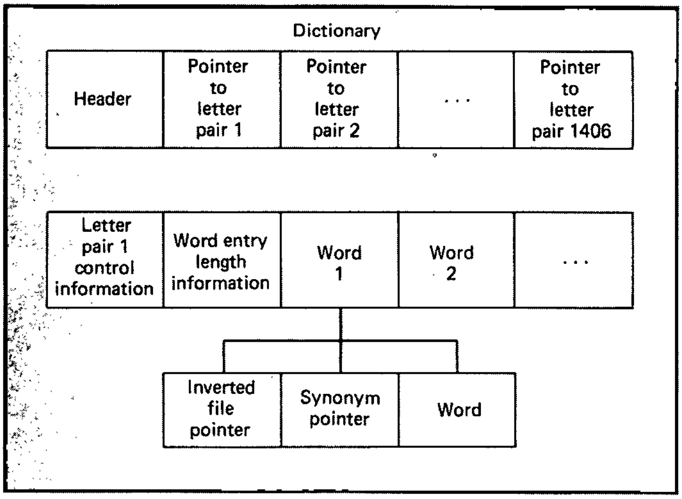

Figure 2-2 STAIRS dictionary organization. (*Adapted from reference 8.*)

the STAIRS system. In that case the location information is not helpful, since the controlled terms are never combined using adjacency or term distance measures.

Documents are found in STAIRS using the SEARCH command. When this command is specified, the SEARCH mode is entered. In this mode the user searches the unformatted portions of the records. Each search statement, normally consisting of a term followed by an operator and another term, is assigned a number of the STAIRS system. For example,

HEART AND DISEASE

is a legitimate search statement in which the normal Boolean AND operator is used. Other legitimate search statements include

HEART
HEART OR DISEASE
HEART NOT DISEASE
HEART$ AND DISEASE$3
(HEART OR CORONARY) AND DISEASE

The "$" symbol indicates truncation and the "$3" indicates truncation with up to three unspecified characters. Other operators of interest are

ADJ

specifying that two terms must be adjacent to one another,

WITH

indicating that the two terms must appear in the same sentence,

SAME

specifying that the two terms must appear in the same paragraph, and

SYN

specifying that the two terms are to be considered as synonyms. The use of these operators is based on a straightforward manipulation of the files presented in Fig. 2-1. Adjacency (ADJ), same sentence (WITH), and same paragraph (SAME) operations use the available location information to ensure that the necessary retrieval criteria have been met. For example, a search statement such as

PROGRAMMING WITH LANGUAGE

specifies that the two terms must appear in the same sentence.

The synonym dictionary is developed by recognizing term equivalences specified by a SYN command. Thus two terms are considered as synonymous when they are connected by the SYN operator. If COMPUTER and MACHINE are both entries in the existing dictionary,

COMPUTER SYN MACHINE

means that the user needs to specify only one of the terms to formulate a query covering either term. Note that a place is left in the dictionary for the synonym pointers. The SYN operation can easily be implemented by placing in the synonym pointer location of each component of a synonymous pair the location of the other component. A query which specifies COMPUTER as a term would then automatically also refer to (be pointed to) the term MACHINE. Document reference numbers for both terms would be retrieved using either term in a query.

The following hierarchy of operations is used to process the STAIRS statements:

```
ADJ
    SYN
         WITH
               AND, NOT
                           OR, XOR
```

with the ADJ performed first, and OR or XOR (exclusive OR) operations performed last. The XOR operator indicates that a document is selected whenever it contains either of the specified terms but not both. Parentheses are allowed in order to alter the order in which the operations are performed.

The DIALOG and STAIRS operations appear reasonably similar even though the commands actually used are named differently and the underlying file structures differ in many details.

To manipulate structured data, the STAIRS system uses the completely separate SELECT mode of operation. In that mode the formatted fields of data normally containing the values of record attributes are searchable. The SELECT mode provides new search operators and operates on portions of the documents different from those specified earlier for the SEARCH mode. For instance, a special AND operation is provided which acts across several distinct formatted fields (major AND); another AND operator is used to operate within a given field. The same is true for the OR operator. For example, the query

SEX EQ MALE AND HAIR EQ (BROWN OR BLOND)

selects records for which the HAIR field is specified as BROWN or BLOND and at the same time the SEX field equals MALE.

In the SELECT mode special relational operators may be used in a query statement in addition to the standard search operators. These relational operators specify restrictions on the values of certain attributes attached to the documents. For instance, one may wish to retrieve all items described by an attribute called AGE with a value for AGE greater than 30; alternatively all items may be wanted with an attribute called SEX equal to FEMALE. In the SELECT mode each attribute is given a name and the attributes are characterized by particular values in each given document. In other words, a given attribute called AGE may be defined in the data base and the value associated with that attribute may be set equal to 25 in a specific document. A query statement used to retrieve this document might specify

AGE EQ 25

where EQ represents an operator specifying equality. Other relational operations include not equal (NE), not greater than (NG), not less than (NL), greater than (GT), less than (LT), within limits (WL), and outside limits (OL). A query statement such as

SEX EQ MALE AND AGE WL 25 , 35

specifies that the items to be retrieved must have attributes called SEX and AGE, the value of the SEX attribute being equal to "MALE," and the value of the AGE attribute lying within the specified limits of 25 to 35. As in the SEARCH mode, the system assigns numbers to the individual query statements.

Although the search operators may not be mixed between the SELECT and the SEARCH modes, one can use a query number from the SELECT mode as a part of a query statement in the SEARCH mode. Thus, if the earlier query including the attributes SEX and AGE is assigned query statement number 1 by the system, then the corresponding set of document reference numbers can be combined with other sets based on the specification of words in the SEARCH mode. For example, in the SEARCH mode one might use a query statement such as

1 AND INFORMATION ADJ SPECIALIST

to indicate that the items to be retrieved must have all the following properties:

1 The formatted attribute SEX must be equal to "MALE."

2 The formatted attribute AGE must have a value within the limits of 25 and 35.

3 The term INFORMATION must appear in the item.

4 The term SPECIALIST must appear in the item.

5 The term INFORMATION must immediately precede the term SPECIALIST in the item.

Note that the adjacency operator (INFORMATION ADJ SPECIALIST) is evaluated before the AND operator in accordance with the precedence previously established.

The processing of information in the SELECT mode differs from the typical information retrieval processing. Rather it fits into a framework explicitly used to handle structured information. The IMS (Information Management System) data base management system is often associated with the STAIRS retrieval system to carry out the structured data manipulations. IMS uses inverted file structures in the sense that for each value of each allowable attribute a list of associated document reference numbers is stored by the system. Processing of the AND and OR operators is therefore a matter of set intersection and set union as previously described. To handle the remaining operators such as "greater than" and "less than" it is necessary to determine if the specified attribute values meet the desired relational criterion.

Another feature of STAIRS is its ability to rank the retrieved documents according to one of several prespecified algorithms. The RANK command is used to operate in a special ranking mode. In the RANK mode the initial search is carried out using the SEARCH mode of operations previously described—sets developed in the SELECT mode are not available in the RANK mode. Once a document set has been chosen for retrieval using the SEARCH mode operations, a ranking of the documents can be obtained by assigning values or weights to the individual terms associated with a document. The retrieved documents may then be presented to the user in ranked order according to the sum of the weights of all terms that match the terms included in the user's query. The value of a term associated with a document is determined by a user selected combination of:

1 The frequency of the term in the document
2 The frequency of the term in the retrieved set
3 The number of documents in the retrieved set in which the term occurs

One particular term weighting algorithm uses the following formula:

$$\text{Value of term} = \frac{\begin{array}{c}\text{frequency of}\\ \text{the term in}\\ \text{the document}\end{array} \times \begin{array}{c}\text{frequency of}\\ \text{the term in}\\ \text{the retrieved set}\end{array}}{\begin{array}{c}\text{number of documents in retrieved}\\ \text{set containing the term}\end{array}}$$

Consider, for example, a term such as SALINE which appears in 152 distinct documents retrieved by a particular query for a total of 1,247 times in all. Assuming that this term occurs 16 times in a particular document, the value of the term SALINE for this document is then determined as

$$\text{Value of term} = \frac{16 \times 1{,}247}{152} = 131.26$$

A final value can be calculated for each document by summing the values of all terms which match the query terms. In the RANK mode, documents are presented to the user in decreasing order of the term value function, the assumption being that this order corresponds to a decreasing order of presumed relevance of the documents with respect to the corresponding query.

STAIRS is a uniquely powerful system. It is designed to offer a great deal of flexibility in the sense that it offers all the power of a free text search system in addition to the formatted retrieval capability based on the values of specific attribute fields. However, the STAIRS system may be expensive to use in that it requires a data base management system in addition to the retrieval system, as well as a large storage capability for the term location, frequency, synonym, and associated information. Furthermore, the user must have a large IBM computer system available for use.

C The Bibliographic Retrieval Services (BRS) System

The Bibliographic Retrieval Services (BRS) system is a system that uses STAIRS as a basis [9]. BRS is a commercially available system operating on about forty data bases. The system had its origins in the biomedical communications network of the State University of New York.

BRS operates exclusively as an information retrieval system and not as a data base management system. Thus, many of the commands included in STAIRS are eliminated. Since no data base management facility is available to the user, there is no need to distinguish between the SELECT and the SEARCH modes. The user may use a LIMIT operator, however, to specify values such as publication year or language. In addition the ranking ability is not present in BRS. Hence the RANK mode has also been deleted. The one remaining mode of operation, the SEARCH mode requires no special identification. The operations used are those included in the STAIRS SEARCH mode with certain added features. One specific extension allows the qualification of a search statement after the search has already been conducted. Thus, one can supply special field designations in the query statements that limit the output produced by an earlier search. For instance, the code TI refers to the title field of a document. A search statement such as

```
1: PROGRAMMING ADJ LANG$
```

first identifies all documents containing the term PROGRAMMING adjacent to terms which begin with the characters LANG (such as LANGUAGE or LANGUAGES). The number 1: designates the query number. The initial search statement can be modified by specifying

```
2: 1.TI.
```

which implies that search statement 1 is to be modified to restrict the two search terms to the title field only. If a document includes the terms in adjacent positions in the abstract of the document but not in the title, then that document will not be retrieved.

An advantage that BRS maintains over the STAIRS system is its simplicity and processing efficiency. The removal of many of the commands, modes, and functional requirements available in STAIRS produces a much cleaner and simpler processing framework. Fewer decisions must be made and simpler programs are used. Furthermore, users do not need either their own computer or their own data bases, and the system is easier to learn to use than STAIRS. The designers of BRS have thus identified the most used portions of STAIRS and confined their attention to the implementation of a streamlined information retrieval system.

D The MEDLARS System

Perhaps the most famous of all available information retrieval systems is the MEDLARS system of the National Library of Medicine (NLM). This system was built as a result of activities initiated in 1964 aimed at the publication of an automated form of Index Medicus. Experiments with on-line bibliographic retrieval systems began in 1967. The first system used the Abridged Index Medicus (AIM) as a data base that became accessible through the Teletypewriter Exchange Network (TWX). AIM-TWX proved the viability of an on-line information retrieval system.

Following the construction of the early system NLM in cooperation with the System Development Corporation (SDC) modified the on-line retrieval of bibliographic information—timeshared (ORBIT) system to meet the special needs of NLM. MEDLARS first appeared in 1971, and several revisions have since been made both to the data base structure (MEDLARS) and to the on-line search package (ELHILL). The number of available data bases has increased and the capabilities of the system have been enhanced [10–12]. Unlike the other systems previously mentioned, the coverage of MEDLARS is largely restricted to documents in the biomedical area.

The structure of the MEDLARS system is principally based on the use of inverted files. Three different files actually constitute this system: the INDEX file, the POSTINGS file, and the DATA file. The data file stores the complete information associated with each record; this includes all the data which can be displayed for the user. Each record is identified by a unique reference number, in this case called the computer assigned number (CAN).

The index file contains all the unique search terms, such as author names, terms from a controlled vocabulary, numbers such as dates, and classification codes. Each entry in the index specifies the term and the document field in which it may be found. For instance, JONES (AU) specifies that the term JONES is located in the author field of the document. Following this information, a two-part number is recorded in the index. The first portion of this number is a reference to the postings file. That is, it designates the position in the postings file where information about this term begins. The second portion of the number specifies the number of postings associated with this term. Table

Table 2-2 MEDLARS Inverted Index Organization

Search term	Sequence number address	Postings
ABDOMEN (MH)	527	3213
AGED (MH)	1073	51604
AGEE JW (AU)	1075	4
⋮	⋮	⋮
ZYMOSAN (MH)	10379	62

Adapted from reference 11.

2-2 shows the organization of the index file. The term ABDOMEN occurs in the main heading (MH) field of the given document, information about the term begins at location 527 in the postings file, and 3,213 different postings are associated with the term. The postings file contains the CAN numbers which identify the specific documents associated with each term. Associated with the term ABDOMEN, 3,213 distinct CAN numbers are therefore listed beginning at location 527. Figure 2-3 describes the overall structure of the MEDLARS system and the path of a query through the various modules of the system.

The commands used by the MEDLARS system are similar to those described earlier for the DIALOG and BRS systems. The format differs in that no search mode is selected by the user and no parentheses are allowed. The user must adhere strictly to a hierarchy which requires that all AND operations are performed prior to any OR operation. Thus if a user is interested in "computer languages" or "programming languages" then

```
1: COMPUTER OR PROGRAMMING AND LANGUAGE
```

will not produce the desired result. Rather, the two statements

```
1:COMPUTER OR PROGRAMMING
2:1 AND LANGUAGE
```

are necessary.

The user is allowed to conduct a string search within MEDLARS. A preliminary search is first conducted and the retrieved set of documents is then scanned to find a specific string of characters in the texts. Thus, a given set of documents can be examined on a letter-by-letter basis in the hope of detecting a given word or phrase occurrence. For instance, the command

```
STRINGSEARCH 1:HEART DI:
```

can be used to find those records identified as a result of a previously defined

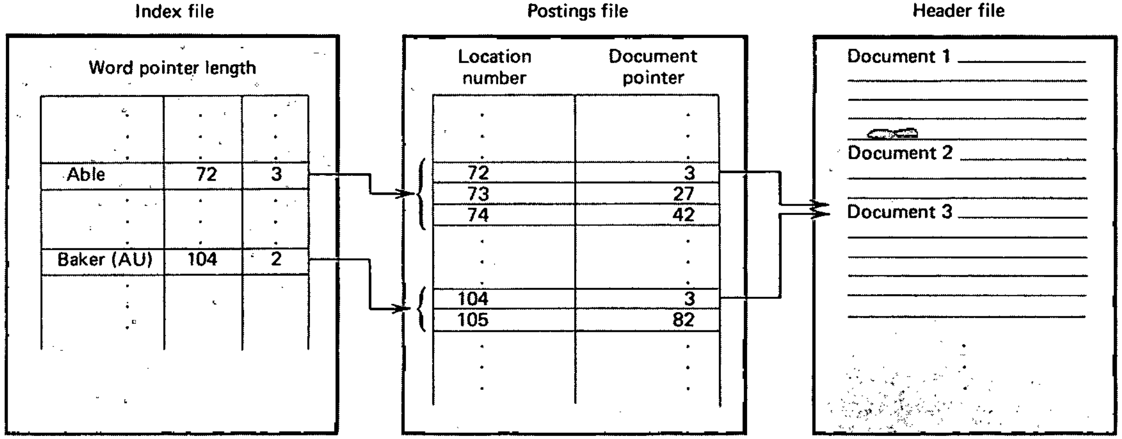

Figure 2-3 MEDLARS file organization. A search statement ABLE AND BAKER would first identify document identifier sets [3 27 42] and [3 82]; the logical AND operator would then be used to indicate document 3 as the only item to be retrieved. (*Adapted from reference 11.*)

search statement 1 which exactly match the string of characters HEART DI. The user may then be increasingly confident that the search results are indeed related to the concept of heart disease. Note that the query HEART AND DISEASE may well retrieve items that have nothing directly to do with heart disease, such as, for example, items dealing with the effects of lung disease on the heart. On the other hand, documents containing the terms "HEART DIMENSIONS" will also satisfy the sample string search.

A string search may be expensive to perform when conducted by most available algorithms. The MEDLARS system therefore limits the potential cost by restricting the string search to document sets retrieved by earlier searches, and by allocating limited time slices for the string search operation. At the end of a given time slice, the system informs the user of the number of documents found so far containing the requested character string. The user is then asked to approve the continuation of the string search process. The same procedure is followed at the end of every time slice. The user may be quickly discouraged by the repetitive procedure. In fact, most sophisticated users limit string searches to small sets of documents.

The efficiency and effectiveness of the MEDLARS system have been repeatedly evaluated. Because the system constitutes the first international online information retrieval system it enjoys a worldwide reputation. There will be further occasions to refer to this system later in this text.

E The ORBIT System

The ORBIT system provided the basic foundation for the design of MEDLARS. It should not be surprising that the two systems exhibit many common features. The basic operational environment is common in both systems. The existing differences are related to the specific procedures designed to help the user in the search process. For instance ORBIT allows the user to generate a chronological display of various search statements entered by the user during the search effort. MEDLARS does not include such a feature. Both MEDLARS and ORBIT do, however, allow the user to view a hierarchical diagram of the search logic [13,14].

F The Information Bank

The Information Bank provides access to the news articles and editorials published in *The New York Times,* and to additional articles from numerous other publications. The search vocabulary is strictly controlled, and the logical operators AND, OR, and NOT are used to relate the terms included in a search statement. Terms entered by a user for which an entry is not found in the term index initiate a dialogue between system and user. The user is eventually asked to select terms from a displayed list of available terms. Searches may be restricted by specifying fields such as date or type of material. The output may be sorted chronologically.

The Information Bank is powerful because the available data base constitutes a unique source of newspaper materials which is not otherwise easily accessible. Plans are under way, however, to render the data base of the Information Bank also accessible through the BRS system. The Information Bank may be particularly valuable to public figures, university professors, students, and in fact to anybody concerned with current or past public events. The Information Bank significantly contributes to the current popularity of on-line information retrieval systems. On the other hand, the technical design features of the system are essentially the same as those of the other systems discussed earlier [15,16,17].

G The LEXIS System

The LEXIS system offered by Mead Data Central provides a service specifically devoted to the manipulation of legal information. The documents are indexed automatically from the text of the items which is stored in its entirety in the system. The searcher uses the traditional Boolean logic to connect the text words included in the search statements. In addition, word location information may be used by insisting that terms appear in a specific portion of a document, within a specified number of words of one another, and in a particular word order. The search terms consist of text words assumed to have a specific meaning related to the document content. Common words such as "the," "it," and "her" are excluded. The common terms are deleted at the time the document is first entered into the system and are not processed any further or entered in the inverted file. Hyphenated words such as ANTI-TRUST are used as two separate terms in the LEXIS system. Thus, the term DATA-BASE is stored as two entries under DATA and BASE, and is considered distinct from DATABASE. Some special word endings can be recognized by LEXIS and reduced to common forms; for example, the terms CITY, CITIES, CITY'S, and CITIES' all retrieve the same document set. Complex morphological differences such as CHILD and its plural CHILDREN are not, however, recognized as equivalent terms.

The processing of terms necessary to remove common words and to isolate terms with common word stems occurs prior to the use of the inverted file. Thus, the LEXIS input processing is distinctly different from the document input and analysis methods used by the other systems described so far. No other publicly available system stores the full text of all documents and uses it for search and retrieval on the scale of LEXIS. Following the elimination of common words and the generation of word stems, LEXIS uses an inverted file system similar to those included in the other retrieval systems. The inverted file retains word location information which may be used either with or without specified word order information. In this sense, the LEXIS system performs much like DIALOG or STAIRS [18].

4 ENHANCEMENTS OF BASIC RETRIEVAL STRATEGY

The use of an inverted file structure appears to be a prominent feature of the existing commercial information retrieval systems. The main inverted file concepts appear to have been used initially on a computing device by Herman Hollerith, who introduced punched cards to handle the computational work for the 1890 United States Census. Since that time the basic inverted file design has not been radically modified. New term location and term frequency information has been added, as in the STAIRS, DIALOG, and LEXIS systems. But the file organization using individual terms and document reference numbers has remained unchanged. This organization can easily handle Boolean operators by translating the AND, OR, and NOT operations into the set intersection, set union, and set difference, respectively. Thus, inverted file procedures are easy to implement and offer the user a powerful way of expressing information needs.

The use of unweighted term combinations such as

ALPHA AND BETA

implies that the user considers the terms ALPHA and BETA equally important. If these terms are not equally important, some means is required to express term importance. Typically the searcher is asked to specify term importance by assigning numeric values to terms. For instance, values between 1 and 10 could be used to designate terms of little importance (1) as well as terms of most importance (10). The user may also be asked to specify a threshold to determine which documents are to be retrieved and which are to be rejected. Thus, given the weights and thresholds

ALPHA = 4
BETA = 5
GAMMA = 6
Threshold = 10

a query such as

ALPHA OR BETA OR GAMMA

would retrieve the following document sets in order:

1 Documents containing terms ALPHA and BETA and GAMMA (4 + 5 + 6 ≥ 10)
2 Documents containing the terms BETA and GAMMA (5 + 6 ≥ 10)
3 Documents containing the terms ALPHA and GAMMA (4 + 6 ≥ 10)

This assumes that the terms occurring in the documents are unweighted and that satisfactory methods are available for choosing appropriate weights for the query terms.

The query term weights produce a partial ranking of the retrieved documents which may improve user satisfaction. However, in practical situations term weighting has not produced substantial improvements in search satisfaction. The reason seems to be that a great deal of effort is required on the part of the searcher to determine correct weights and assign them to the various terms. Thus, de facto, most searches are conducted using the basic inverted file system organizations with unweighted terms.

The inverted file process permits a great deal of flexibility in the design of the user-system interfaces. Even though most systems are based on the same fundamental file organization and search strategy, different ways exist for presenting the information to the user. The manner in which commands are presented, the hierarchy of operations, and the different operators allowed in each system render each system unique.

Currently, several hundred data bases are associated with the various systems. These data bases include millions of documents. Thus, an enormous investment exists in systems designed to operate with inverted files. The introduction of changes and modifications to the existing commercial retrieval systems must therefore depend upon not only the technical feasibility but also the economic impact of the alterations. The great expansion in the availability and use of retrieval systems over the last dozen years leads one to expect a continued growth in system development and services in the years to come.

REFERENCES

[1] D. Fife, K. Rankin, E. Feng, J. Walder, and B. Marron, A Technical Index of Interactive Information Systems, Systems and Software Division, Institute for Computer Science and Technology, National Bureau of Standards, Washington, D.C., March 1974.

[2] M.E. Senko, File Organization and Management Information Systems, Chapter 4, Annual Review of Information Science and Technology, C. Cuadra, editor, Vol. 4, Encyclopaedia Britannica, Chicago, Illinois, 1969, pp. 111–143.

[3] F.W. Lancaster, On-Line Information Systems, Encyclopedia of Library and Information Science, Vol. 20, Marcel Dekker, New York, 1977.

[4] F.W. Lancaster and E.G. Fayen, Information Retrieval On-Line, Melville Publishing Co., Los Angeles, California, 1973.

[5] D. Lefkowitz, File Structures for On-Line Systems, Spartan Books, Rochelle Park, New York, 1964.

[6] C. Bourne and B. Anderson, DIALOG Lab Workbook, 2nd Edition, Lockheed Information Systems, Palo Alto, California, 1979.

[7] DIALOG Information Retrieval Service, Database Catalog, Lockheed Missiles and Space Co., Inc., Palo Alto, California, May 1980.

[8] IBM World Trade Corporation, IBM System/370 (OS/VS), Storage and Information Retrieval System/Vertical Storage (STAIRS/VS) Reference Manual.

[9] Bibliographic Retrieval Services, Inc., BRS Bulletin, Schenectady, New York.

[10] National Library of Medicine, On-Line Services Reference Manual, U.S. Department of Health, Education and Welfare, Bethesda, Maryland, January 1980.

[11] National Library of Medicine, MEDLARS, The Computerized Literature Retrieval Services of the National Library of Medicine, Department of Health, Education and Welfare, Publication NIH 79-1286, January 1979.

[12] D.B. McCarn, MEDLINE: An Introduction to On-Line Searching, Journal of the American Society for Information Science, Vol. 31, No. 3, May 1980, pp. 181–192.
[13] System Development Corporation, Highlights and Hints: ORBIT, System Development Corporation Search, Santa Monica, California.
[14] System Development Corporation, Search Service News, Newsletter of the System Development Corporation Search Service, Santa Monica, California.
[15] J. Rothman, The Times Information Bank on Campus, EDUCOM Bulletin, Vol. 8, No. 3, Fall 1973, pp. 14–19.
[16] The Information Bank, Thesaurus: A Guide for Searching the Information Bank and for Organizing, Cataloging, Indexing, and Searching Collections of Information and Current Events, Parsippany, New Jersey, 1977.
[17] J. Rothman, The New York Times Information Bank, The New York Times, New York, 1969.
[18] Mead Data Central, LEXIS Quick Reference, New York, 1976.

BIBLIOGRAPHIC REMARKS

Readers interested in learning more about the use of inverted files in various areas of application may want to consult the following additional references:

G. Wiederhold, Database Design, Chapter 3, Basic File System Organization, McGraw-Hill Book Company, New York, 1977.
C. Meadow, Applied Data Management, Chapter 3, Data Storage, and Chapter 7, File Maintenance, John Wiley and Sons, Inc., New York, 1976.

For an elementary discussion of data structures see:

G.G. Dodd, Elements of Data Management Systems, Computing Surveys, Vol. 1, No. 2, June 1969, pp. 117–133.

The following items relate inverted file systems specifically to information storage and retrieval:

A.E. Wessel, Computer Aided Information Retrieval, Chapter 9, Computer Software and Some Hardware Considerations, Melville Publishing Company, Los Angeles, California, 1975.
C.J. van Rijsbergen, Information Retrieval, 2nd Edition, Chapter 4, File Structures, Butterworths, London, England, 1979.

The operation of information retrieval systems using inverted files is described in the following texts:

F.W. Lancaster and E.G. Fayen, Information Retrieval On-Line, Melville Publishing Company, Los Angeles, California, 1973.
F.W. Lancaster, Information Retrieval Systems: Characteristics, Testing and Evaluation, 2nd Edition, Chapter 2, The Matching Subsystem; Chapter 3, The Application of Computers to Information Retrieval-Off-Line Batch Processing Systems; Chapter 4, On-Line Information Retrieval, John Wiley and Sons, Inc., New York, 1979.

The reader should be careful to distinguish books dealing with file structures from texts covering principally data structures. There is some confusion in terminology between data and file structures and between lists, indexes, and files. In information retrieval, the term data structure normally refers to abstract constructs used to represent the entities and concepts under consideration—for example, documents, terms, and sentences. File structure, on the other hand, refers to the organization of the document files and of the auxiliary files used to access the main document files.

EXERCISES

2-1 Consider the following inverted index

TERM A	1,4,5,6,8
TERM B	2,3,4,6,7,9,10
TERM C	3,5,7,9

Identify the document numbers associated with each of the following retrieval statements

a TERM A AND TERM B
b TERM A OR TERM C
c TERM A OR (TERM B AND TERM C)
d TERM C NOT TERM A

2-2 Using the inverted file structure from Exercise 2-1 describe a procedure which will identify the documents in which TERM A is immediately followed by TERM B.

2-3 If TERM A is assigned a weight of 2 by a particular user, and TERMS B and C are assigned weights of 4 and 3 respectively, which documents will be retrieved by each of the following statements, assuming a retrieval threshold of 6:

a TERM A OR TERM B
b TERM A OR TERM B OR TERM C
c TERM B AND TERM C

2-4 Using any programming language and an arbitrary text string consisting of more than one sentence develop a routine which creates an inverted index for the words from the text and the associated sentence numbers. For example, given the text: "The objects are processed serially. The first of the objects becomes the representative," the appropriate inverted index will be

THE	1	2
OBJECTS	1	2
ARE	1	
PROCESSED	1	
SERIALLY	1	
FIRST	2	
OR	2	
BECOMES	2	
REPRESENTATIVE	2	

Use the program created for Exercise 1-2 to isolate the individual words in the text string.

2-5 Under what circumstances would it be reasonable to keep term location information in an inverted index? Given the example in Exercise 2-4, develop a routine to create an inverted index that includes term location information.

2-6 Create a routine which transforms the inverted index of Exercise 2-4 into alphabetical order.