TEXT RETRIEVAL TECHNIQUES FOR THE AUTOMATED OFFICE

W. Bruce Croft
Mark T. Pezarro*

COINS Technical Report 81-26

October 1981

*Currently at MICOM Co., 5250 Ferrier St., Montreal, Quebec.

Abstract

The advent of the automated office will have far reaching effects. One of its principal consequences will be a dramatic increase in the amount of textual information stored in machine-readable form. The indexing, filing and retrieval of this information will be among the most important functions of office information systems. This paper describes techniques that have been developed for these tasks and how they can be applied to the office environment. The techniques can be implemented efficiently and they provide a flexible interface suitable for casual users.

Introduction

A great deal of attention is currently focussed on office information systems, a new breed of distributed system which will make the entry, editing, filing, analysis and communication of all forms of office information considerably easier, faster, cheaper and more efficient [ELLI80, TSIC 80]. A major research and development effort in this area is underway in both the commercial and academic communities. The generation and editing of text is the subject of research at both XEROX-PARC and MIT in the form of sophisticated systems for document preparation [EHAR 80]. Electronic mail is already a commercial reality with the appearance of systems such as Wang's Mailway or Datapoint's Electronic Mail System. Interestingly enough, however, the problem of retrieving text documents has been somewhat neglected. Clearly this is a vital issue because there is little sense in storing information if it cannot be later found and retrieved.

Current word processing systems are fairly primitive in the area of information management. A leading manufacturer's products do not even allow the user to assign mnemonic names to documents when they are filed. Instead the system assigns a four digit number which the user must remember. The equivalent of this filing system in an office employing manual procedures would be to store the documents in boxes which had labels that had nothing whatsoever to do with their contents. More sophisticated systems such as the Xerox Star [SEYB81], provide electronic "file drawers" and "file folders" in which documents are stored by the user. The burden of finding relevant documents in response to a need for information still rests directly on the user. This does not represent much of an advance over conventional filing systems. We believe that an office information system should be able to provide a filing and retrieval system which is more effective and efficient than any manual system. Moreover, it should have an interface that makes it

easy for casual users to store and locate documents.

This paper discusses some techniques that are appropriate for document filing and retrieval in the office environment. These techniques are based on the statistical analysis of the frequency of occurrence of words in text. The statistical analysis yields sets of keywords (or index terms) used to represent documents and directs the retrieval of documents relevant to a user's query. Much of the early work in this field is described by Salton [SALT68]. More recent developments are covered by Van Rijsbergen [RIJS79].

This statistical approach has advantages over both standard database management techniques and AI techniques. In a database management system, records (in this case, documents) are retrieved when the set of secondary keys (index terms) matches the query specification exactly. In many cases, this method of exact matching is too restrictive for searching documents by their content. Statistical techniques do not insist on an exact match but instead rank the documents according to their similarity to the query. A system which uses AI techniques to produce a complex representation of the text documents by syntactic and semantic analysis would require a database containing general knowledge about the subjects covered in the documents. Setting up and maintaining a knowledge data base involves considerable overheads. The statistical approach has much lower storage and processing requirements and the simple "knowledge database" it uses can be automatically derived from the text of the documents in the system.

The remainder of the paper is devoted to a description of the techniques that could be used in three main areas:

1. indexing text documents
2. retrieving text documents
3. the user interface.

Indexing text documents

Indexing a text document means producing a list of keywords or index terms that describe the document's content. It is similar to the process of creating entries in a library card catalogue for a new book. There will be two types of index terms for a document in the office environment; those index terms describing fixed characteristics of the document such as source, destination and date for a memo and those describing the content of the document as expressed in the text. The terms in the first category will be easy to identify because they are required for every document, whereas those in the second category must be derived from the text automatically or provided manually.

In some systems the person entering the document is expected to provide a list of terms to describe the document's content. There are a number of problems with this approach. Firstly, the person entering the document is probably not the author and therefore may not fully understand its content. This objection could be overcome by making the provision of index terms part of the document writing process. Another more serious problem is the inconsistency of the indexers. Different people will use different words to describe the same subjects. In an environment where people are accessing documents from all over a company or where there are frequent staff turnovers, this will make retrieval difficult. Professional indexing services use indexers trained in the subject area and a set of guidelines for indexing in order to maintain consistency.

Rejection of manual indexing leaves two possibilities - semi-automatic indexing and fully automatic indexing. With semi-automatic indexing, the system would produce a list of candidate index terms which have been derived automatically, along with the draft copy of the text.

The writer of the document would then check the list of terms and add or delete terms as appropriate. In the office environment, the text will normally be proofread and therefore this would appear to be a reasonable method of correcting any gross errors made by the automatic indexing process. A survey of techniques used for automatic indexing appears in Sparck Jones [SPAR74]. A more complicated indexing process based on a statistical model of the occurrence of words in text is described by Harter [HART75].

A simple indexing procedure that should be effective for business text is:

a) Remove all words less than 3 letters long and all special characters.

b) Remove all stopwords (common words such as and, but, the).

c) Remove suffixes to reduce words to common stems.

d) Count occurrences of stems.

It might prove necessary to adjust the above procedure for certain organizations, for example, those which make extensive use of 2-letter acronyms.

This procedure produces a list of index terms (which are the stems) and associated frequency weights. In practice, the word stems will be printed in a more recognizable form (e.g., INDEPENDENT instead of INDEPEND). The frequency weights indicate the importance of the terms in the document. These weights may also be changed by the document writer in the semi-automatic process. A final step in the indexing process may be to remove the lowest weighted terms to reduce the size of the group of stems comprising the document representative. Figure 1 gives an example of a document indexed by this process with all terms of weight 1 removed. It is interesting to note that including phrases in the index terms as well as individual words does not increase the effectiveness of the text retrieval [SALT68]. In fact, Salton shows that other more sophisticated approaches to text analysis are no more effective than this simple procedure.

Title - Evaluation and Selection of File Organizations - A Model and System.
Abstract - This work first discusses the factors that affect file (data base) organization performance, an elusive subject, and then presents a methodology, a model and a programmed system to estimate primarily total storage costs and average access time of several file organizations, given a specific data base, query characterization and device-related specifications. Based on these estimates, an appropriate file structure may be selected for the specific situation. The system is a convenient tool to study file structures and to facilitate as much as possible the process of data base structure design and evaluation.
Journal - CACM September, 1973.
Author - Cardenas, A.F.
Keys - file organization, file structures, file management, file organization performance, file organization model, secondary index organization, simulation, data base, access time, storage requirement, data base analysis, data management.

Document index terms -
file 11 organization 7 data 6 base 6 structure 6
model 3 system 3 access 2 time 2 storage 2
evaluation 2 selection 2 performance 2 management 2
design 2 specific 2

Figure 1 - A document and the derived index terms

One of the major problems with business text is choosing the parts of the text to be used as input to the indexing process. Using the entire text of the document can lead to long processing times and, more critically, large lists of index terms for long documents. Using just the title of a document often does not provide enough information. In a system which indexes scientific documents, such as that shown in Figure 1, the title and abstract are typically used. However, business documents usually do not have abstracts and can even be completely unstructured (i.e., no section headings). Although some investigation is needed of the best methods of indexing business documents, the following guidelines seem reasonable.

a) If the document is short (e.g., a memo), use the entire text.

b) If the document is long, use the title, introductory sections and section headings if they are available.

c) If the document is long and unstructured, use just the title.

In the last category, the system will have to rely more heavily on terms suggested by the writer of the document.

An important part of the indexing process is the removal of stopwords. In a typical document retrieval system there is a list of these stopwords against which the incoming text is checked. In an office information system, this stopword list can be incorporated into the dictionary of words used for spelling correction [PETE80]. This would require a person in the organization to be designated "vocabulary manager". This person's task would be to identify stopwords in the dictionary and flag them as such. A large number of stopwords would be common to every application and thus would be pre-specified.

Another important modification of the spelling dictionary would be to incorporate a thesaurus or synonym dictionary. Experiments have shown that the use of a thesaurus can, in many cases, improve the system performance [SALT68, HARP78]. The thesaurus would also be able to be searched by the users of the system when they are formulating their queries. A thesaurus can be generated automatically [SPAR71], but in the case of the business environment, a more useful thesaurus could be set up by the vocabulary manager.

The implementation of the indexing process described above is straightforward. The major overheads are the stopword list and thesaurus which are incorporated into the spelling dictionary. The processing of the text could even be done while it was being entered, although it is probably better to do the processing when the document is filed on disk. The implementation of the retrieval process is mentioned in the next section.

It should be mentioned that some text retrieval systems (such as the legal information system LEXIS [MEAD75] store and search the full text of the documents. Although full text searching avoids the indexing process, it does so at a cost. In order to obtain reasonable performance, either every term must be put into an inverted file, leading to large storage overheads, or content-addressable hardware [HOLL79] must be used. A more serious criticism is that searching based on matching the text of the query to the text of the document is very restrictive and inflexible. It is really only appropriate for a formalized subset of natural language, such as that used in legal documents.

It has been implicitly assumed up to now that the full text of the document is available in machine-readable form and is stored within the office information system. It is worth noting that the techniques described in this paper could equally well be used to catalogue manually filed documents at the cost of entering the material suggested earlier for the representation of long documents in the indexing process (i.e., title + abstract or title + introductory section(s) + section headings). The full text of the document would be stored manually and the reference to the document stored with the index terms would be some kind of file reference number. Such an approach seems suitable for organizations which have large, existing paper archives or libraries for which the cost of conversion to machine-readable form would be prohibitive.

Retrieving text documents

After the documents have been indexed, they are represented by a set of index terms, possibly with weights attached. Now the problem is to determine which documents are relevant to a query by comparing the query to the document

representatives. Two ways of specifying queries have been used in bibliographic document retrieval systems. The first requires the users to specify their interests using Boolean combinations of index terms (e.g., (WORD and PROCESSOR) or (AUTOMATE and OFFICE)). The documents retrieved are those having representatives which satisfy this Boolean specification. This method is used by the large commercial services (e.g., DIALOG [LOCK76]). The problems with this method are firstly, a heavy burden is placed on the user to produce the correct Boolean expression to retrieve the documents in which he is interested in and, secondly, there are often relevant documents which do not exactly match the query.

The second way of specifying queries is for the user to write a natural language query which is then indexed in the same way as the text of the document. The relevant documents are determined by comparing the set of terms in the query to the sets of terms representing the documents. A great deal of theoretical and experimental work has recently been done on this topic ([ROBE76], [RIJS77], [HARP78], [SPAR79], [CROF79a]). An approach which has been shown to perform well involves ranking the documents in decreasing order of their probability of relevance for a given query. This probability is estimated using the occurrences of terms in the relevant and non-relevant sets of documents for the query. The non-relevant set is approximated very closely by the entire collection of documents; however, the relevant set has not been identified at the start of a search. Initially, therefore, the characterisitics of the relevant set are approximated using the query and the documents are ranked. By using a process known as relevance feedback, better estimates of the relevant set's characteristics are obtained and used to rerank the documents. In relevance feedback, the user is presented with a few of the top documents from the intitial ranking. The user

then identifies the relevant documents amongst those displayed. In this way, a better picture of the user's concept of a relevant document is obtained.

A similar process can take place in a system using Boolean queries. In this case the user has to look at some of the (unranked) retrieved documents, decide how to reformulate the query based on what is contained in those documents and resubmit the query. Relevance feedback using document ranking obviously requires much less effort on the part of the user and, more importantly, it uses the new information in a more effective manner.

In many cases, the user in a business environment is looking for a specific document rather than a group of relevant documents. For these users, it may be useful to combine the Boolean specification of query terms with the ranking of documents. In this way, the user could specify some terms the document must have to be considered and these documents only would be ranked in order of their probabilities of relevance based on the rest of the query. In general, only index terms based on a document's fixed characteristics should be used in this manner. Otherwise, the previously mentioned burden of selecting appropriate content terms would again be placed on the user.

A possible extension of the application of these techniques could be to aid office workers in filtering and ranking incoming documents in machine-readable form. Ackoff [ACK067] observed that a major problem of most managers is an over abundance of irrelevant information. Consequently, "the two most important functions of an information system are **filtration** (or evaluation) and **condensation** ". Automatic condensation of documents is not yet within the realm of everyday practicality but automatic filtering is. A user profile could be specified in natural language for each office worker defining the documents in which he or she was most interested. All incoming documents would then be automatically compared to the profile and ranked according to

their probability of relevance. Documents with a probability of relevance below a certain threshold could be automatically filtered out. It might be desirable to augment the content terms from the user profile with some fixed filtering criteria based on the identity of the sender; for example, no mail from an individual's superior(s) should be filtered out.

The implementation of these retrieval techniques requires a central index for the documents. This index would be required for any retrieval technique and, in a system consisting of a number of stations networked together, the index could be set up in the central file storage. The index would be in the form of an inverted file, where for each term there is a list of the documets indexed by that term. This is another reason for a good stopword list since the less index terms there are in a system, the less storage overhead is involved in the central index. Both the Boolean form of searching and the ranking of documents can be implemented using this inverted file [CROF79b].

The User Interface.

The retrieval methods outlined above allow a very flexible interface to be designed. Three types of query are possible:

a) Boolean combinations of terms. This type would be used for queries involving fixed characteristics of documents (such as source or date) and for specifying compulsory content terms. An example of this type of query is:

```
LIST DOCUMENTS WHERE
     SOURCE = 'SMITH' AND
     DESTINATION = 'JONES' AND
     DATE = 02/11/80
```

This system retrieves all documents written by SMITH to JONES on the 11th February 1980.

b) Natural language specifications of content. These are used to find documents about a particular topic when the fixed characteristics are not known. An example of this type of query would be:

ENTER DESCRIPTION OF DOCUMENTS

> I am interested in correspondence dealing with the purchase of high resolution flicker free displays and matrix printers for word processors.>

After the specification of a natural language query, the system should list the terms derived from the query and encourage the user to indicate some order of importance. Even simple information such as this can lead to significant performance improvements [CROF-79a]. If the user is not satisfied with the terms displayed, extra terms could be added or the thesaurus could be browsed in order to locate more satisfactory terms. After this phase, the system would present a few of the top-ranked documents and the user could either terminate the search or use relevance feedback (simply by specifying the relevance or non-relevance of the documents) to obtain new documents.

c) Example documents. It is possible in this type of system for the user to quote the primary key of a known document and then the system will retrieve documents similar in content to that document. For example:

LIST DOCUMENTS SIMILAR TO DOC. 17345

The rest of the search procedure is the same as described for queries of type b.

These three types of query would be integrated into a powerful but simple interface.

Conclusion

It is not enough to provide users of office information systems with sophisticated tools for creating, filing, communicating and analyzing information. The problem of retrieving relevant information from the vast body of machine-readable information that these systems will make available must also be addressed. This paper has outlined a number of techniques resulting from recent research in information retrieval which offer the possibility of developing text retrieval facilities which are both powerful and easy to use. The use of these techniques promises a way of bringing the information explosion at least partially under control.

References

ACKO67 Ackoff, R.L. "Management misinformation systems", **Management Science, 14/4:** (1967).

CROF79a Croft, W.B. and Harper, D.J. "Using probabilistic models of document retrieval without relevance information." **Journal of Documentation, 35:** 285-295; 1979.

CROF79b Croft, W.B. "On the implementation of some models of document retrieval." Proceedings of the 2nd International ACM SIGIR Conference, **SIGIR Forum, 15:** 71-77; 1979.

EHAR80 Ehardt, J.L. and Seybold, P.B. "Experimental systems: Xerox Document System, M.I.T. Etude." **The Seybold Report on Word Processing 3/9,** Seybold Publications, Box 644, Media, PA 18063 (1980).

ELLI80 Ellis, C.A. and Nutt, G.J. "Office information systems and computer science", **ACM Computing Surveys, 12/1:** 27-60 (1980).

HARP78 Harper, D.J. and Van Rijsbergen, C.J. "An evaluation of feedback in document retrieval using co-occurrence data." **Journal of Documentation, 34:** 189-216; 1978.

HART75 Harter, S.P. "A probabilistic approach to automatic keyword indexing. Part I: On the distribution of specialty words in a technical literature. Part II: An algorithm for probabilistic indexing." **Journal of the American Society for Information Science, 26:** 197-206 and 280-289; 1975.

HOLL79 Hollaar, L.A. "Unconventional computer architectures for information retrieval. In M.E. Williams (Ed.)", **Annual Review of Information Science and Technology, 14.** Knowledge Industry Publications, White Plains (1979).

LOCK76 Lockheed Information Systems. **A brief guide to DIALOG searching.** Palo Alto, California (1976).

MEAD75 Mead Data Central, Inc. **LEXIS: A primer.** New York (1975).

PETE80 Peterson, J.L. "Computer programs for detecting and correcting spelling errors". **Comm. ACM 23/12,** 676-687; (1980).

RIJS77 Van Rijsbergen, C.J. "A theoretical basis for the use of co-occurrence data in information retrieval." **Journal of Documentation, 33:** 106-119; 1977.

RIJS79 Van Rijsbergen, C.J. **Information Retrieval.** 2nd Edition, Butterworths, London (1979).

ROBE76 Robertson, S.E. and Sparck Jones, K. "Relevance weighting of search terms." **Journal of the American Society of Information Science, 27:** 129-146; 1976.

SALT68 Salton, G. **Automatic Information Organization and Retrieval.** McGraw-Hill, New York (1968).

SEYB81 Seybold, J. "The Xerox Star: A Professional Workstation", **The Seybold Report on Word Processing 4/5,** Seybold Publications, Box 644, Media, PA 18063 (1981).

SPAR71 Sparck Jones, K. **Automatic Keyword Classification for Information Retrieval.** Butterworths, London (1971).

SPAR74 Sparck Jones, K. "Automatic Indexing." **Journal of Documentation, 30:** 393-432; 1974.

SPAR79 Sparck Jones, K. "Experiments in relevance weighting of search terms." **Information Processing and Management, 15:** 133-144; 1979.

TSIC80 Tsichritzis, D.L. and Lochovsky, F.H. "Office information systems: Challenge for the 80s." **Proc. IEEE, 68/9:** 1054-1059 (1980).