ORGANIZATION OF A RETROSPECTIVE DOCUMENT RETRIEVAL SYSTEM BASED ON FRAGMENTS (L'ORGANISATION D'UN SYSTÈME RÉTROSPECTIF POUR L'ACCÈS DE DONNÉS D'UNE BANQUE D'INFORMATION BASÉ SUR DES FRAGMENTS)

Ernst Schuegraf
Department of Mathematics
St. Francis Xavier University
Antigonish, N.S. B0H 1C0

ABSTRACT

A description of a retrospective document retrieval system is given in which equifrequent fragments are used as language elements for compression coding and retrieval. The critical parameters governing the system are identified and restrictions imposed by the parameters on the selection of fragments are outlined. Changes necessary in the processing of a query are mentioned and two conditions are postulated for the successful operation of a fragment based retrieval system. (On décrit un système rétrospectif pour l'accès de donnés d'une banque d'information dans lequel on emploie des fragments de fréquence uniforme pour le code de compression et la recherche (l'accès). On identifie les paramètres importants qui gouvernent de système et on souligne les limites imposés par les paramètres quant au choix de fragments. On mentionne les changements nécessaires pour introduire une interrogation et on propose deux conditions pour bien réussir dans l'opération d'un système d'accès (de recherche) basé sur des fragments.)

INTRODUCTION

The discovery of the computers potential for the solution of non-numeric problems and its subsequent application to information processing has produced considerable benefits to information scientists as well as to the users of information. An example of this beneficial effect is the availability of computerized search services of machine readable document data bases, which free the scientist from the tedious task of extensive literature searches. The customer of a search service submits a set of questions called his profile, which represents his current interest. The profile consists of a set of index terms and logical connectors.

Two distinctly different types of search services can be identified. The most common one is the current awareness or SDI search, in which the user profiles are compared against a small data base, which is replaced at regular intervals. This fixed set of profiles retrieves up-to-date references to new appearing documents from this changing data base, to

keep the user informed on current research. In contrast to the abundance of SDI searches is the rarity of retrospective retrieval services. These services provide searches on a large static data base which has accumulated over a long period of time. However, there is no fixed set of user profiles, but rather a rapidly changing set of questions, which when answered, will not be used again on the same data base. The rarity of these retrospective systems can be partially explained by the complexity of the necessary computer programs and the cost of manipulating large volumes of data by the computer. Methods suitable for SDI searches are impractical and uneconomical for retrospective systems. The differences in the operation of the two services necessitate a different organization for retrospective retrieval systems.

SYSTEM STRUCTURE

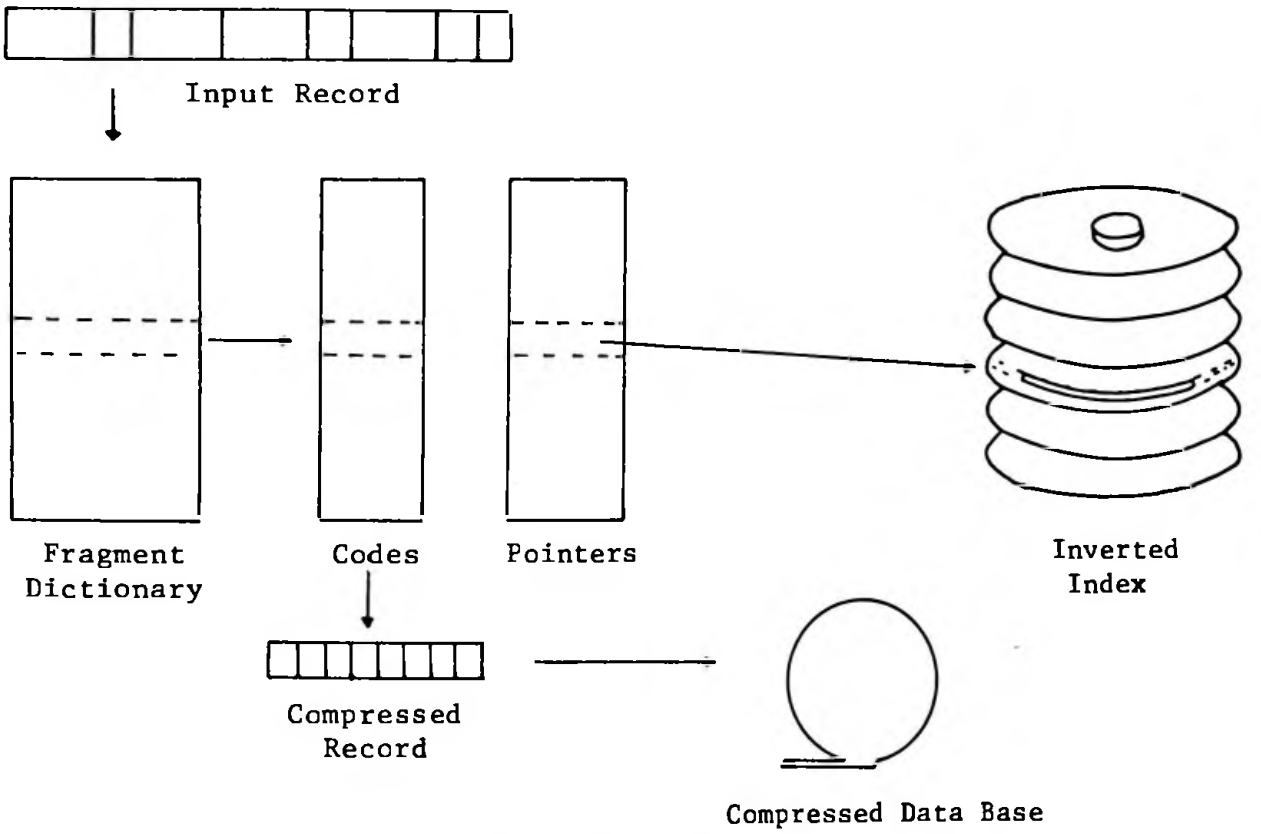
Retrospective retrieval systems encounter two major problems: the large amount of data to be manipulated and the problem of getting a fast response to a query. The latter problem is normally solved by creating an auxiliary file called an inverted index. This index is produced when the system is generated. The elements used for indexing are also utilized as language elements for data base compression, which partially solves the problem arising from large volume of data. A detailed description of a retrospective document retrieval system based on words as language elements is provided by Thiel and Heaps (1972), while a mathematical analysis of the compression scheme used in the system is given by Heaps (1972).

Considerable problems arise when words are chosen as the language elements (Schuegraf and Heaps 1973). It is therefore proposed to use a suggestion by Clare et al. (1972) and switch the language elements. Most of the problems arising in the organization of word oriented systems can be traced to the hyperbolic distribution of word frequencies, known as Zipf's law (Zipf 1949). The new suggested indexing and compression elements are called fragments, and are supposed to have the property of equifrequency. A fragment may be defined as a variable length character string. Word-fragments are strings completely contained in words and text fragments are those embedded in a record. The latter may contain blanks and punctuation symbols, while the former may not.

Example: Word-fragments "MATION", "ICAL", "AUTO", "PORT".
Text Fragments "AND THE F", "MATION.", "EVER, T".

An ordered dictionary of these fragments limited to a fixed size is the major component of a retrospective retrieval system based on fragments. Its role in the system and its relation with the inverted file and the compressed data base are shown in Figure 1.

RETROSPECTIVE DOCUMENT RETRIEVAL



SYSTEM STRUCTURE

For each element in the dictionary there is an associated list of entries in the inverted file. This list includes the identification number of all the documents, which contain the particular index string. The inverted index is put into action in the retrieval phase of the process for determining the desired document numbers. The complete documents are stored in the compressed data base without loss of information by storing codes for the dictionary fragments found in the record. The code represents the position of the entry in the dictionary subject to a previously agreed order of elements. At the time the system is built, all fragments are identified which are embedded in the record and which when concatenated represent the whole record. At the same time the entries in the inverted index are made and the compressed data base is generated by storing the fixed length codes for the selected dictionary elements. During system operation the inverted file is used to determine by application of specified logical operations (AND, OR, NOT) the document numbers of hits. These documents are retrieved from the compressed data base, decoded by substituting the character string for the code, and output to the user.

SYSTEM PARAMETERS

The organization of the system as described above is subject to the influences of many parameters, but four are of major significance. These are the number of elements in the fragment dictionary ND, and their average fragment length AF. The relation between the total number of characters in the data base TC and the average frequency F of occurrence of a fragment is given by

$$TC = ND * AF * F \qquad (1)$$

The number of entries in the inverted file is given by

$$I = TC/AF, \qquad (2)$$

and the length L of the compressed data base by

$$L = (TC/AF) * \lceil \log_2 (ND) \qquad (3)$$

One of the objectives in the design of the system is to limit the size of the dictionary, so that during decoding the dictionary may be stored in the available computer memory. The size of the dictionary in characters is given by

$$S = ND * AF \qquad (4)$$

For reasons of efficient coding a threshold frequency is introduced and all fragments with a frequency below the threshold are not considered as possible elements for compression and indexing. The threshold in effect selects the average frequency F. The equifrequency property of the dictionary fragments may be measured quantitatively by the "efficiency" E given by

$$E = -\sum_{1}^{ND} p(i) \log_2 p(i) / \log_2 (ND)$$

with p(i) being the probability of the i^{th} dictionary element. This measure is closely related to the efficiency of a coding scheme and is one for a uniform distribution and less than one for any other.

It is evident from an inspection of equations (1)-(3) that the size of the systems' components is a function of the average fragment length. This fact together with other aspects related to compression restrict the selection of dictionary fragments. They can be summarized as

1) The set of dictionary fragments must be complete in the sense that every document citation can be represented by concatenating dictionary elements;

2) The dictionary set should maximize the average fragment length in order to minimize the storage requirements for the inverted file and the compressed data base;

3) The selected set of fragments should maximize the efficiency E, which is equivalent to requiring approximate equifrequency;

4) The dictionary set should not be over redundant in the sense of having unnecessary overlap between its members. In addition it should not be possible to choose a smaller set that satisfies requirements 1-3.

It has been shown (Schuegraf and Heaps 1973) that a suitable algorithm can be developed to satisfy those requirements. Experiments with various thresholds and samples of different sizes have been performed, and it was found (Schuegraf 1974) that the threshold is the only parameter governing dictionary size and average fragments length. The higher the threshold frequency, the smaller the dictionary set and the average fragment length. Problems encountered in the compression phase when using fragments as the language elements are discussed in a further paper (Schuegraf and Heaps 1975), and different compression algorithms are compared. The following title compressed with only 9 different elements from a text fragment dictionary may serve as an example for a compressed record.

"|T|ALE|S|-FROM-|THE-S|K|IP|P|ER.|"

QUERY PROCESSING

Special procedures are required to process a query in a system that uses fragments as index elements. The first step is the determination of all fragments contained or overlapping into the given search terms. This may be achieved by an algorithm utilizing some auxiliary tables (Schuegraf 1974) which can handle regular as well as truncated search terms. It is sufficient to deal only with the implementation of the three logical operators AND, OR and NOT. The first two are implemented in regular fashion by locating the index fragments in the dictionary and retrieving with the aid of the pointers the associated rows of document numbers of the inverted index. The proper logical operations on these rows are then carried out. For AND this consists of taking the intersection of the two lists (i.e. keeping only document numbers common to both lists), while for OR it is the union of the two lists. The problem arises with the NOT operator which in word oriented systems is implemented by creating a new list which contains all the document numbers of the first list, which do not appear on the list of the negated term.

This procedure is unacceptable in a fragment system, since the same fragment may occur more than once in a query as part of different search terms. In one case the index fragment may be negated and in the other it

may not. Furthermore one index fragment may be a part of two or more index terms, and all documents containing that index fragment would be excluded a priori; a fact which may reduce the recall considerably.

The solution to this problem is to ignore the negated search terms during the search phase with the inverted index. However, to eliminate the documents containing negated terms we must subject all documents to a sequential search for the full logic after they have been decoded.

A second problem arises with all those search terms which are concatenation of single letter fragments. Since it is impractical to generate index entries for single letter fragments, no index entries exist and the term cannot be retrieved. This phenomenon is not serious, as long as other terms in the query contain index fragments. However, in this case we must also subject the decoded documents to a sequential search for the full logic. The problem becomes more difficult, if all the search terms are concatenation of single letters. The system can process only those queries which contain at least one index fragment. We may, therefore, postulate that two conditions are necessary for the operations of a fragment based retrospective retrieval system.

A query must contain at least one index fragment, and a sequential search for the full logic is necessary on all documents retrieved by the index. This will not only eliminate the problem with the neglected NOT operator, but will also assure that no documents are retrieved because of spurious coincidence of index fragments.

Experiments have shown (Fokker and Lynch 1974) that fragments produce satisfactory results when applied to the retrieval of author names. Additional results with regard to retrieval performance are given by Creasey et al. (1974) and they stated that further tests in an operational environment will be carried out in the near future.

REFERENCES

BARTON, I.J., CREASEY, S.E., LYNCH, M.F., SNELL, M.J., 1974 An information theoretic approach to text searching in direct access systems. Comm. Assoc. Comp. Mach. Vol. 17, 345-350.

CLARE, A.C., COOK, E.M., LYNCH, M.F., 1972 The identification of variable length equifrequent character strings in a natural language database. Computer Jr. Vol. 15, 259-262.

FOKKER, D.W., LYNCH, M.F., 1974 Application of the variety generator approach to searches of personal names in bibliographic data bases. Part 1/Part 2. Jr. Libr. Automation Vol. 7, 105-118, 201-213.

HEAPS, H.S. 1972 Storage analysis of a compression coding for document data bases. INFOR Vol. 10, 47-61.

SCHUEGRAF, E.J., HEAPS, H.S. 1973 Selection of equifrequent word fragments for information retrieval. Infor. Stor. Retr. Vol. 9, 697-711.

_____, HEAPS, H.S. 1975 A comparison of algorithms for data base compression by use of fragments as language elements. Infor. Stor. Retr. Vol. 11 (in press).

SCHUEGRAF, E.J. 1974 The use of equifrequent fragments in retrospective retrieval systems. Ph.D. Thesis, Dept. of Computing Science, University of Alberta, Edmonton (TR 74-11)

THIEL, L.H., HEAPS, H.J. 1972 Program design for retrospective searches on large data bases. Inform. Stor. Retr. Vol. 6, 137-153.

ZIPF, C.K. 1949 Human Behaviour and the Principle of Least Effort. Addison Wesley, Cambridge Mass.