KEEPING UP WITH THE LITERATURE

A PRACTICAL EXAMPLE OF RETRIEVING INFORMATION

J.A. Hall
Research Laboratory
British Museum
Great Russell St.
London, W.C.1

Abstract

This paper is a description of the computer-based system used at the British Museum Research Laboratory for storage and retrieval of references to the technical literature of art and archaeology. The system, developed both to satisfy the internal needs of the laboratory, and to be used in the wider context of 'Art and Archaeology Technical Abstracts', is based on the use of keywords for describing subject matter. An efficient file structure is used, permitting fast retrieval. The structure and use of the system are outlined.

1 Introduction

The aim of this paper is twofold: to give an outline of the structure and use of the information retrieval system developed at the British Museum Research Laboratory for literature references, and thereby to illustrate some of the possibilities for quite advanced and efficient systems on small computers.

The first aim is important, because anyone who uses 'Art and Archaeology Technical Abstracts (AATA) will indirectly use this system, and anyone concerned with this field may find that direct access to the system would be useful. To put this in context, a brief history of the project might help: The system was originally requested by various members of the B.M. laboratory staff in early 1974; it was felt that the computer could be used as a finding and indexing tool for all the various articles, offprints, pamphlets etc. which were read or kept either by the laboratory library or by individual members of staff. We developed the system on this basis but it was clear from the start that if it worked well it could be of most use in conjunction with AATA, the abstract journal covering the relevant field. We therefore approached AATA and asked them to provide us with data for the system by passing on to us their abstracts, together with keywords to provide a brief description of the content of each cited work. In return we offer use of the system for retrieval to anyone interested, and we use the system to produce the annual index to AATA. The main bulk of our data, therefore, consists of references scanned for AATA since about the beginning of 1975, but in addition there is a body of more local data on offprints, unpublished work etc., held in the laboratory.

The second aim of the paper is to illustrate the sort of approach which can be applied to designing quite powerful systems using a relatively small computer. We are using a minicomputer - a Hewlett Packard 2100 - which is used in the Research Laboratory for a variety of real time and other laboratory tasks. The operating system allows us only about 24000 characters of core, and at present we have one 2½ million character cartridge disc devoted to the literature reference system. Within these limitations we could store about 20000 references and give retrieval time of a few seconds.

2 Methodology

2.1 Keywords

The fundamental requirement at the inception of the system was a method of accessing the literature by subject matter. The first problem, therefore, was to decide how to describe the subject matter of an article. There are many possible approaches, of which the most obvious are:

1 Using the title - basing a KWIC or similar index on the text of the title.
2 Using a free text abstract and, e.g., a KWIC index.
3 A method based on explicit keyword descriptors.

Method 1 was ruled out on the grounds that the title is notoriously a very poor indicator of the content of the work. Methods 2 and 3 both suffer from the fact that someone has to read the article to find out what it is about; however, considering the relatively low volume of accessions and also the fact that this was already done for AATA, this did not seem to be a disadvantage.

Method 2 has the advantage over 3 that the text of the abstract puts each keyword in a context; however, it is uneconomical of storage and vocabulary control is very difficult. Furthermore, searching would be slow unless very complicated retrieval techniques were used.

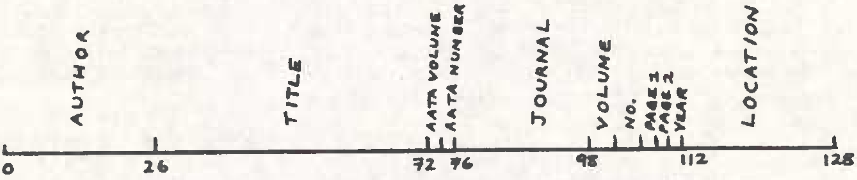
Method 3 was chosen, largely on the grounds that it is easy to devise extremely efficient storage structures and search algorithms for keyword retrieval. It also has the advantage of making vocabulary control very simple. Although this causes problems, we feel that a system where these problems are identified at the input stage offers a more realistic approach than an uncontrolled system where the problem would go unrecognised - the non-retrieval of items can easily not be noticed!

The major problem with this approach is the potential for "false drops" on retrieval, because the keyword context is not recorded. A reference on vases, discussing the painting of vases, and their conservation, would have the keywords "vase", "painting", "conservation" and would be found in a search on painting conservation. However, serious false drops do seem to be rare, partly because of the fairly restricted nature of the field covered.

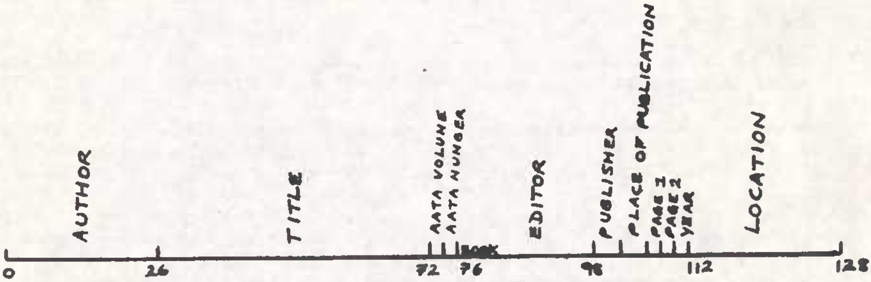
2.2 The Storage Structure

Storing the bibliographic data - author, journal, year and so on - is straightforward. We had available a package for data storage and retrieval, based on the simple idea of dividing a fixed length record into fixed, named, fields. Figure 1 shows the structure of the bibliographic records, with the character positions within the record indicated. Obviously, here there is a compromise between completeness and economy of storage. Journal abbreviations, publisher and place abbreviations are all standardised and separate files kept of the abbreviations and their meanings, to try and ensure consistency. Often titles must be abbreviated, and here there is some judgement needed to achieve meaningful abbreviations.

This has not solved the problem of storing the keywords. Unfortunately, the 'obvious' solution, of storing with each reference a list of the relevant keywords, is totally unsatisfactory. One would have to look through each reference to find if it contained a particular keyword - or, indeed, to find if the keyword even existed.

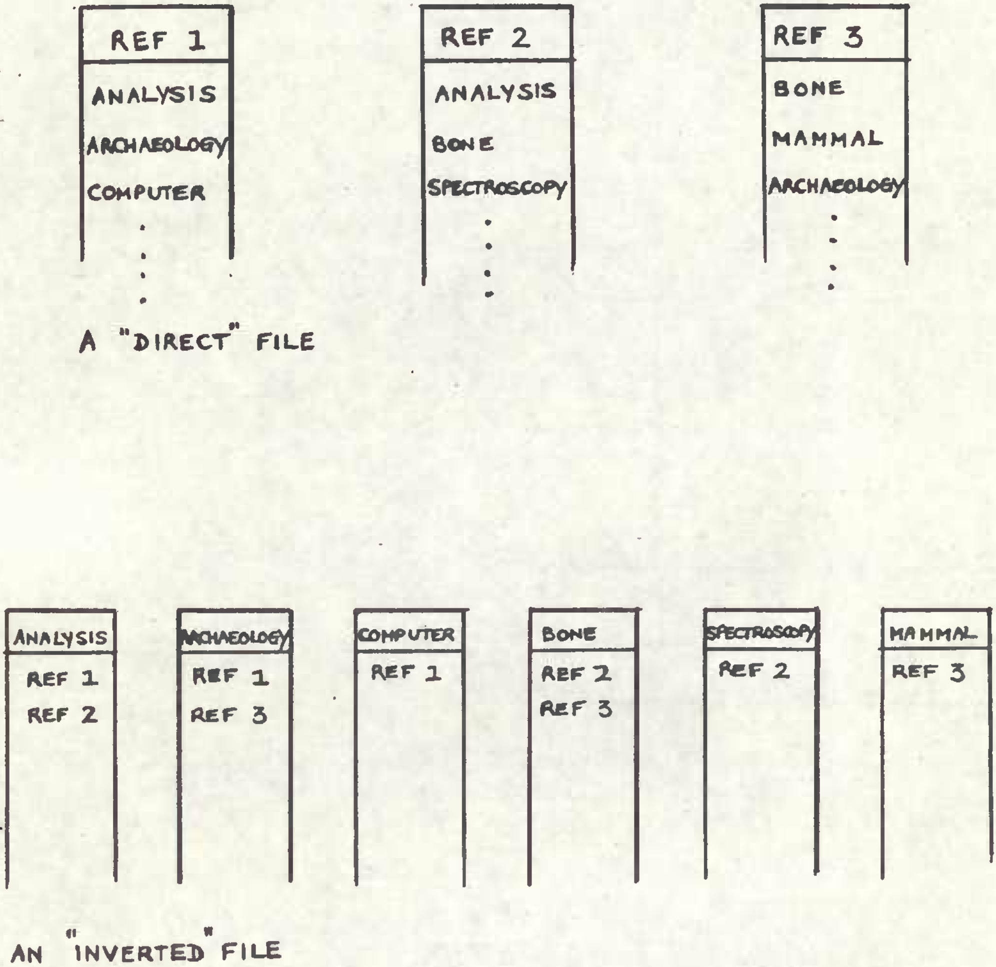


FORMAT FOR JOURNAL ARTICLES



FORMAT FOR BOOKS

FIGURE 1 : FORMAT OF BIBLIOGRAPHIC DATA



A "DIRECT" FILE

AN "INVERTED" FILE

FIGURE 2: DIRECT AND INVERTED FILES

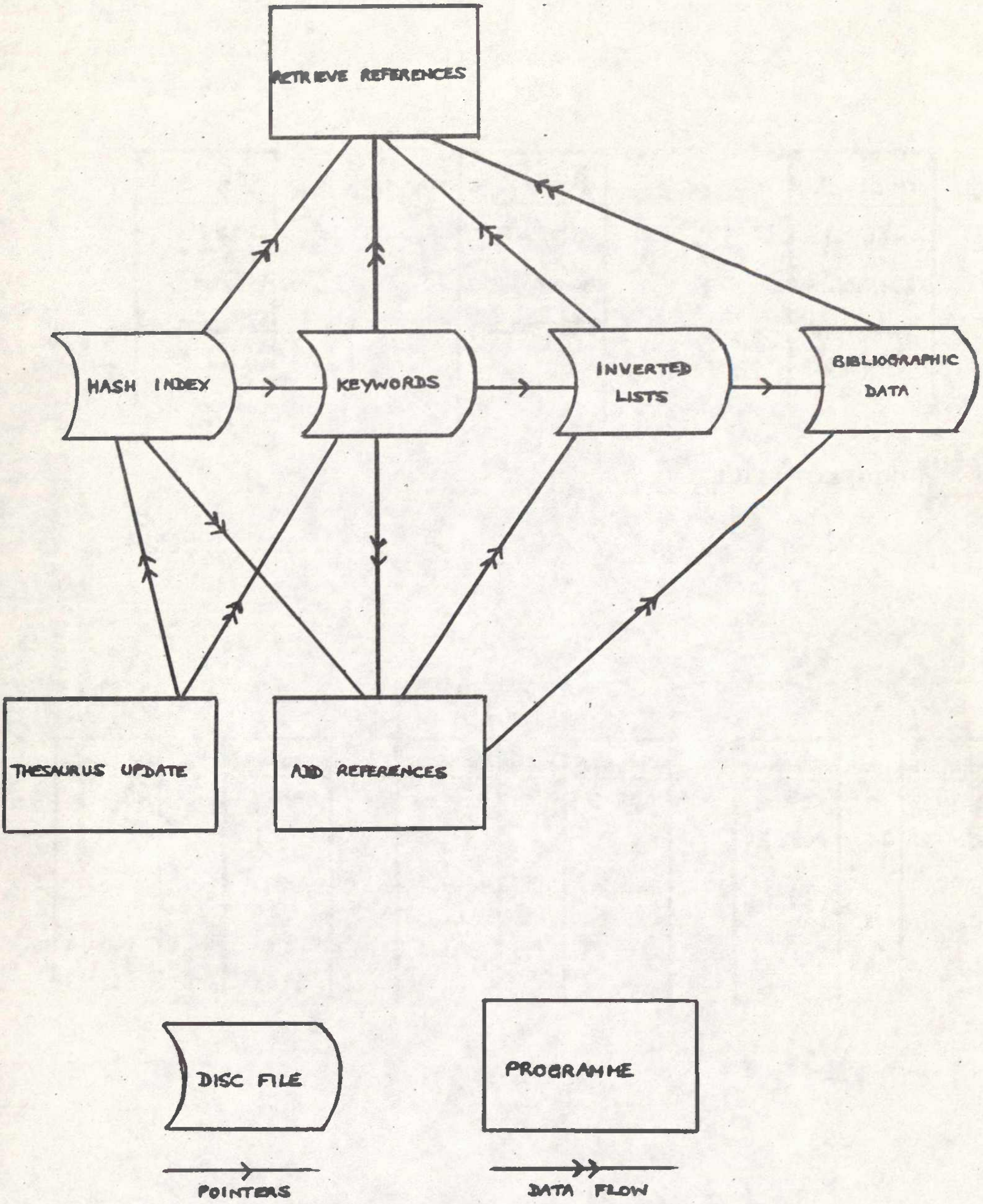


FIGURE 3 : THE OVERALL SYSTEM

Furthermore, we would either have to limit the number of keywords per reference, or use a complicated variable-length record structure. Instead, we separate the keywords completely from the bibliographic files, and use the well known concept of 'inverted files'. An inverted file, illustrated in figure 2, contains, rather than a set of references each with keywords, a set of keywords each with its references. There are many advantages in this arrangement:

1 Vocabulary control is simple - the list of keywords is much shorter than the complete file would be.
2 There is considerable space saving. The entries in the inverted file are simply record numbers, taking only two characters each, so the amount of storage for the keywords and lists is only of the order $n(l+2m)$ (where n is the number of keywords, l the length of a keyword and m the number of references per keyword) instead of nlm.
3 The structure is relatively simple, since although overflow records must be provided for the inverted file this is very simply done.
4 Retrieval is very efficient. A search on subject matter can be performed without reference to the bibliographic file at all - merely be manipulating lists of record numbers, almost any retrieval condition can be statisfied.

There is one further refinement needed to make access as fast as possible. Since the keyword file will be built up in the order that keywords are encountered, it will have to be scanned completely to check each keyword. To overcome this we index the keywords using the hash technique as this is both very fast and makes updating very straightforward.

Figure 3 shows the complete system in simplified form. Access to the keywords is via the hash index, and the keywords in turn give access to the bibliographic information via the inverted lists. Notice that the main input programme (ADD REFERENCES) only reads the index and keyword files - it does not update them. Adding new keywords can only be done by the THESAURUS programme which can only be used by someone who knows the correct security code.

3 Operation of the System

3.1 Input

Input is typed in a fixed format either on to paper tape or directly into the computer. The format used is based on the AATA abstract forms, and these may be used as input documents. The system checks the keywords for validity, and if the input is satisfactory it is added to the files. At the same time, a print out of the new data is produced, giving the bibliographic data, the keywords, and the system accession number of that reference. This can be proof read, and it is then filed in accession number order to form the accession list.

3.2 Retrieval

The primary means of retrieval is by keyword. The user types in a list of keywords connected by "AND", "OR", or "NOT". The retrieval programme uses the index, the keyword file and the inverted lists, but not the bibliographic data, to produce a list of references satisfying the request. This list consists of the accession numbers of the references, and can be displayed if required, so that a user with aocess to the accession list can retrieve the references even if the bibliographic file is not available. If the bibliographic file is on line, however, further selection on such criteria as author, date of publication eto. can be done. The results can be displayed at the terminal or printed, and either full references or accession numbers may be requested.

3.3 Indexing

The same techniques are used to produce the annual index to AATA. Here, each index term, of which there are currently about 8000, is a combination of keywords. The index terms are first sorted and then standard retrieval routines used to select the relevant references and print the AATA abstract numbers.