Personal Librarian: a Tool for the Literature Classroom

D. S. MIALL
University of Alberta, Canada

Abstract

A text retrieval program for IBM-compatible microcomputers, *Personal Librarian* (formerly known as SIRE), is reviewed for its relevance to undergraduate study of literary texts in the classroom. In addition to supporting the standard text retrieval functions, with searches for words and collocates, support of Boolean operators, and a proximity operator for collocates, retrieved documents are rank ordered according to their estimated relevance to the query. *Personal Librarian* also allows searches that use a whole document as a query term and an Expand command which produces a thesaurus of collocates unique to the text base being studied. In this way the program enables the user to progress from word occurrences to the level of themes and ideas. These facilities will make the software particularly useful in supporting classroom discussion of texts, where suggestiveness and a fairly intuitive set of search pathways are more important than close linguistic analysis.

In introducing students of English and other literatures to the computer as a medium for the analysis of texts, there is a need for computer software that will enable rapid searching and retrieval of a text base. A range of text retrieval packages has been created, and many are now available for a microcomputer, but so far few have found a regular place in the undergraduate curriculum. For classroom use, where the focus of attention is on the content of a text rather than on its linguistic or computational dimensions, there is need for a program that is simple to understand and use, but powerful enough to enable different search strategies to be implemented and the results of searches to be viewed immediately on screen. *Personal Librarian*, a program available for IBM-compatible microcomputers, meets this basic need; it also provides some facilities to support text retrieval which are unique, and specially suited to work in the humanities classroom.

Until recently the only program explicitly designed to meet such a need was *WordCruncher*: a brief account of the advantages of *WordCruncher* will help to set *Personal Librarian* in context. Almost no prior training is required to make effective use of *WordCruncher* at a first attempt. A student introduced to the program can choose a text (or 'book') from the 'bookshelf', shown when the program is first called up, can go from there to a screen containing an alphabetical list of all the words in the text, and by choosing a word can call up all instances of the use of the word in a window where the search word is highlighted. Given that students approach a text with some understanding, with questions they wish to ask, *WordCruncher* provides a highly efficient tool for focusing on the vocabulary of a text. The program is flexible enough to support many different types of enquiry: in addition to searching on single words, words can be searched for in a variety of combinations; the distribution of words across a text base can be reviewed; and a concordance or index can quickly be compiled and printed out. With these facilities, *WordCruncher* provides a tool offering immediate support to classroom discussions of literature or to students' work on their own projects.

At a basic level *Personal Librarian* offers the same facility: the user can immediately find all occurrences of a given word, or search for two or more words in combination. Beyond this, however, *Personal Librarian*, while lacking the word distribution and concordancing facilities of *WordCruncher*, does offer two methods which are particularly fruitful for the study of literary texts. First, on completing a search, the program will rank order the sections of a text that it has retrieved; and, secondly, the program has a routine for automatically locating significant co-occurrences of a given search word. These methods, which no other microcomputer text retrieval package offers, can, if used judiciously, give the user a better grasp of the conceptual structure of a text: they enable searches to reach beyond the level of words towards themes and ideas. In this respect *Personal Librarian* puts the computer closer to the level at which debate about a text normally takes place in the literature classroom, and it gives the student a tool which will stimulate exploration and discovery.

I now look in more detail at the operation of the program, its rationale and history, and note some of its limitations. I will deal primarily with the DOS-based version of the program: more recently a version operating under MS-WINDOWS has been released which offers some additional advantages. A version for the Macintosh is also said to be imminent.

Personal Librarian derives from an earlier research project at Syracuse University on text data bases, in which the primary aim was to develop a method for automatically assessing the relevance of documents to a search query. In standard information retrieval, as in most data bases, a document is retrieved if it matches the terms in the search query, and documents will be displayed in the order in which they occur in the raw data base. In *Personal Librarian* a weight is given to the search terms in a retrieved document, based on the number of times the terms occur in the document and on the length of the document. The documents are then rank ordered, so that those documents that are most relevant to the query will be seen first. The algorithms for this were developed during the 1970s in an earlier version of the program called SIRE.

A study carried out to help validate the ranking method, involved asking physicists to frame statements

Correspondence: David S. Miall, Department of English, University of Alberta, Edmonton, Alberta, Canada, T6G 2E5.

based on their research interests; search terms derived from the statements were then used to retrieve citations and abstracts of articles. The physicists rated the retrieved articles on how relevant they were to their interests; the resulting rank orders correlated at a highly significant level with the rankings produced by the program (Noreault *et al.*, 1977). The search procedure thus provides a more efficient and rapid identification of relevant documents.

The ranking of documents is the most immediate and attractive feature of the program: it requires no action from the user. On starting the program the user is presented with a screen showing a menu of options on the top line, and a bar at the bottom of the screen where the user enters commands. There are no other menus. All commands can be entered by a single letter, followed by the relevant arguments. The most recent search query is shown below the command line. The normal mode of operation of *Personal Librarian*, however, is the search, and this requires only the entry of search terms. Thus a single word entered on the command line is enough to initiate a search. For more complex searches Boolean operators are supported, multiple levels of brackets, right-hand truncation (called 'stemming' in the language of *Personal Librarian*), and wildcards. If the user is searching for collocates the distance of the words can be controlled by the proximity operator: thus 'sleep w/20 dreams' will locate occurrences of the target words within a span of twenty words.

The standard result of a search is a list of the documents occurring first in rank order: the first two lines of one or more of the fields of each document is shown, together with a note of the total number of documents retrieved at the foot of the screen. An example list is shown in Fig. 1, derived from the poetry of Coleridge. Following the list each document itself can be viewed on screen: nineteen lines are shown at once, with the search word or words highlighted (see Fig. 2); paging through the document, if it consists of more than nineteen lines, is done by pressing the Enter key. Pressing the down or up arrow keys will jump to the next or previous document in rank order. An alternative to the rank order list is the bar chart (see Fig. 3): a histogram of estimated relevance is shown, in which the first document in the rank order scores 100% and subsequent documents are scaled accordingly. This provides a useful impression of the effect of the search command. The bar chart shows immediately if the search command was too broad or too narrow and should be issued again in modified form.

Searching is extremely rapid since, like *WordCruncher*, texts are pre-indexed. The size of the index can be controlled by use of a stopword list, which can be written by the user (a default list is provided with the program, but this will sometimes be too extensive for literary purposes). *Personal Librarian* enables documents to be organized by fields, as in a data base: up to 256 fields per document (or record) can be specified, and field lengths are unlimited. This makes it suitable for a number of the requirements of literary text analysis, although for a straight literary text base one field per document may be all that is required (for example, each poem, or section of a longer poem, in a collection of poetry would form one document). The field names are inserted as ordinary text, taking up one line and enclosed in hyphens, for instance: '-Author-'. The end of a document is marked by '-end-'. The use of the hyphen as field name delimiter is a minor irritation, since literary texts often include a dash or hyphen at line endings; but a word processor can be used to find these and add another character outside each hyphen to 'protect' it.

Taking a specific example, I found that indexing a file 665K in length, the complete poetry of Coleridge, on an 80286 computer with a hard disk took 12 minutes (the poetry was divided into documents consisting of one field). By contrast, *WordCruncher*, which admittedly requires three levels of coding, took 55 minutes to index the same material. Retrieval is almost immediate: within 4 seconds of entering a one word search term the first nine document titles are listed on screen. A more complex search adds another 1 or 2 seconds. Obtaining the full text of the opening of the first document then takes

```
Get Display Jump > < Quit Help Set List Bar Alpha Expand Past SOrt EDit Write
 Rank
  1) title: Christabel  SECTION 7
    ¦L145  Outside her kennel, the mastiff old
  2) title: Dejection: An Ode  SECTION 7
           But hush! there is a pause of deepest silence!
  3) title: An Effusion at Evening  SECTION 2
           Propitious Fancy hears the votive sigh --
  4) title: Elegy: Imitatated from one of Akenside's Blank-verse Inscriptions
    ¦Pg.69
  5) title: Songs of the Pixies  SECTION 4
           When Evening's dusky car
  6) title: Happiness  SECTION 4
           'Tis thine the converse deep to hold
  7) title: Pain
    ¦Pg.17
  8) title: Lines: On an Autumnal Evening  SECTION 2
           O dear Deceit! I see the Maiden rise,
  9) title: To Lord Stanhope on Reading his Late Protest in the House of Lord
    ¦Pg.89

Query 6        Retrieved 67
Enter command>
SLEEP* OR ASLEEP                                    DB: \pl\coleridg\coleridg
```

Fig. 1. Rank ordered list of documents retrieved.

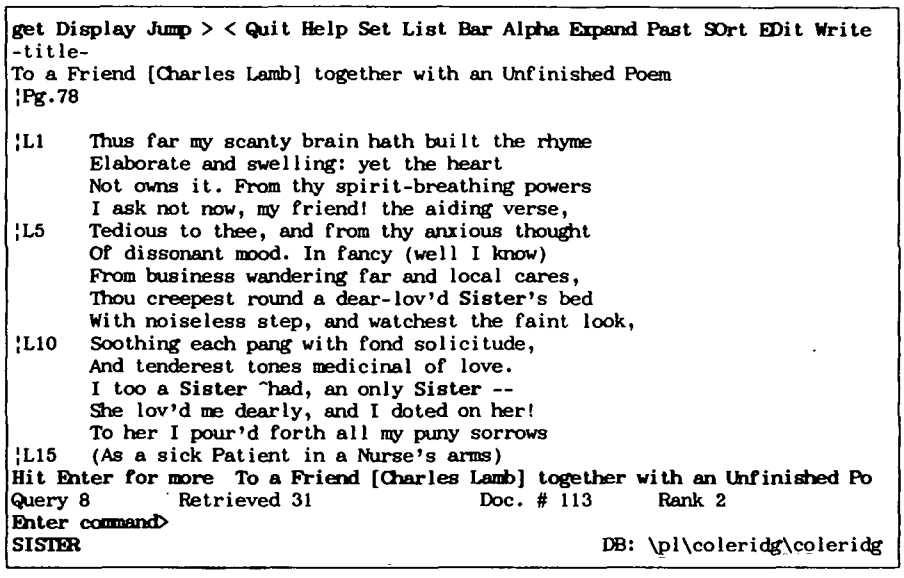

Fig. 2. Displaying retrieved text, with search words highlighted.

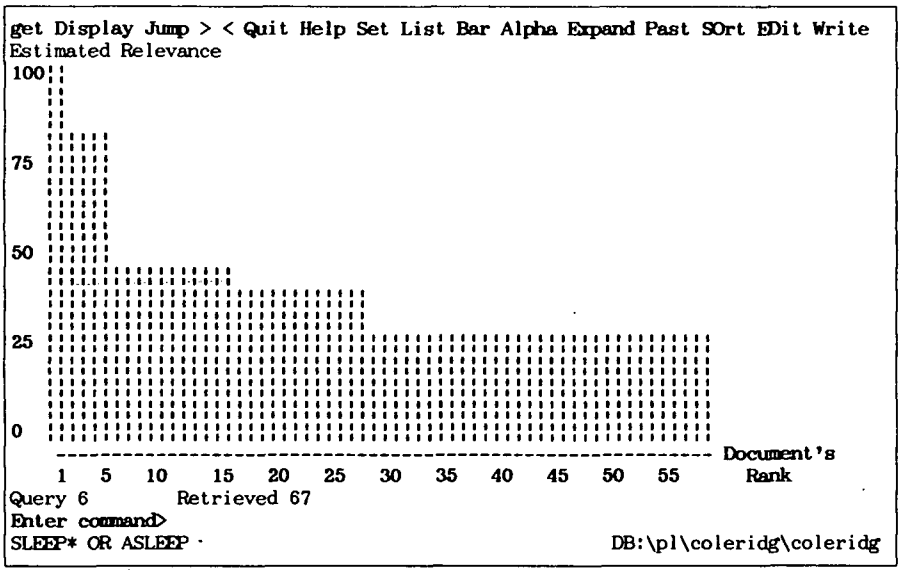

Fig. 3. Bar chart showing relevance of retrieved documents.

only another second, and paging of additional documents takes place with no perceptible pause. During retrieval the list can be recovered by pressing 'l', and the bar chart can be seen by pressing 'b'. A document can also be seen out of rank order by entering 'd' and the rank order number, eg. 'd 23'. It is also possible to page through documents outside the retrieval list by using the left or right arrow keys to move backwards or forwards in the text base from a given document.

More sophisticated searches can be performed by using a document itself as a search term. This has the effect of making all the significant words in the document into search words. Documents similar to the target document will be retrieved and rank ordered; each will show some degree of commonality in its themes and concerns. This search takes longer to complete: using a Coleridge poem of twenty-three lines as the search term, *Personal Librarian* required 30 seconds before producing the rank order list. As a classroom technique for widening the range of poems or other documents students will consider, however, this facility has considerable potential.

The most unusual aspect of *Personal Librarian* is its 'Expand' facility. The program will search for significant collocates to a given word or set of words, and produce a list of forty-eight words on screen. Again, this search takes time: length of time appears to depend in part on how many occurrences of the target word or words *Personal Librarian* has to take into account. The Expand command on one word, that occurs seventeen times in the Coleridge file, took 117 seconds to produce the list of collocates. Expanding on this word and one other word, a co-occurrence of which there are only two examples, took only 25 seconds. The results of the expand command, however, can often be illuminating. The effect is a type of thesaurus entry for the target word or words,

unique to the text base being consulted. As such it can often reveal special properties of the vocabulary and assocations of the author under study. For example, Coleridge lost his only sister at an early age: this event is referred to in several poems, and in other poems the idea of sisterhood holds a special place for Coleridge. The Expand command with 'sister' as the key word indicated typical Coleridgean concerns: *gloomy*, *distress*, *woes*, *pain*, and *agony*, but also a number of occurrences of love, shown here in its 'stem' form, *lov* (see Fig. 4).

Again, the Expand facility has considerable potential for focusing and extending students' researches on a text base. Unfortunately, the authors of *Personal Librarian* provide no information about the Expand algorithm. It is not clear on what basis the collocates are selected, and the list shown on screen does not seem to occur in a systematic order; a collocate occurring sixteen times may be listed next to one occurring only twice. Thus the results of an Expand command should be treated with some caution.

In the DOS version an Expand list will disappear from screen as soon as another command is issued. In the WINDOWS version the list can be kept on screen or recalled as a reference point while the collocates it shows are examined in more detail (this is also true of the other displays offered by Personal Librarian, such as the bar chart). Several other options are available from the menu at the top of the screen. The Alpha command, issued with a target word, shows nineteen lines of the dictionary of words in the text, centring on the target word and including the number of occurrences (it is similar to the 'Select Word' screen of *WordCruncher*). The Set command shows a summary of the current default options, for example whether the list or the bar chart is shown first following a search. Previous queries are remembered by *Personal Librarian*, and these can be listed on screen by issuing the Past command. This is helpful if the user wants to repeat a previous query, add another term to a query, or combine two queries, since previous queries can be run again merely by typing 'Q _ ' and the number of the query. A set of retrieved documents can be sorted by entering the Sort command, together with the name of the field on which the documents are to be sorted.

Personal Librarian also provides several methods for extracting documents if the user wants to print out the results of a search or edit the material with a word processor. Specified documents can be dumped to a file, called DOCOUT, which can be edited through a word processor after leaving *Personal Librarian*. Alternatively the Write command, which is toggled on and off by pressing 'w', will dump the current document on screen to the DOCOUT file. In addition, an Edit command is provided which suspends *Personal Librarian* and calls up whatever word processor has been specified in the installation routine. On exiting the word processor, *Personal Librarian* resumes at the point where it was left.

In summary, *Personal Librarian* provides an efficient context for searching large text bases, with a fairly intuitive set of pathways to assist exploration following an initial search. Simple searches can be carried out with little knowledge or training. An on-line help facility provides some guidance for the more elaborate searches and other commands. My version came with a small sample file called Movies, which was used as the training example in the manual. The manual itself, provided in an A4 ring back binder, is divided into two main sections: a beginner's guide and reference, and an 'administrator's guide' which shows how to prepare a file for Personal Librarian and how to index it. The tutorial section assumes no knowledge of data bases and provides a useful introduction to the program. I found some inaccuracies in the manual: the syntax of some of the commands appears to have changed since the manual was last printed. Like most commercially produced manuals, however, this one is inappropriate for use with humanities students, who require a tutorial built around a more relevant set of examples.

The Coleridge file I installed with *Personal Librarian*, which was 665K in size, took a total of 560K for the twelve other index and system files needed by *Personal*

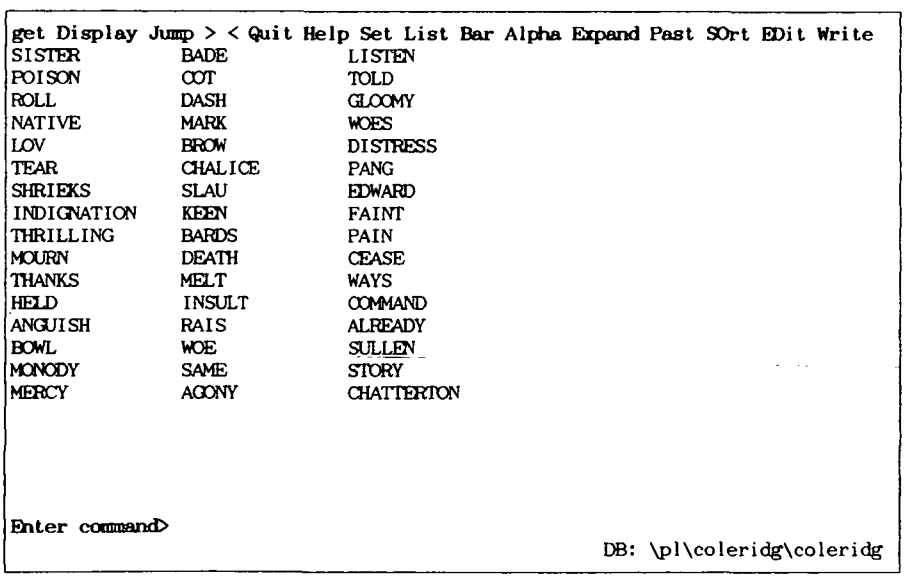

Fig. 4. Expand list for 'sister'.

Librarian. Larger files with more redundancy would probably need a lower proportion of additional disk space (the developers quote an overhead of 55% in discussing the earlier program, SIRE: Koll, *et al.*, 1984). *WordCruncher* took marginally more space to index the same file. Users of *Personal Librarian* should note, however, that during the indexing process a temporary file is created which appears to be at least twice the size of the source file, thus a large margin of empty disk space will be needed when indexing a new file.

In several respects *Personal Librarian* offers a more productive environment for studying texts than its closest rival, *WordCruncher*. The search by document, and the Expand facility, both provide suggestive ways of tracing further documents at the level of themes and ideas, which will often be of value in the literature classroom. Since the basis on which Personal Librarian undertakes an Expand search is unclear, detailed study of collocations will still require a scholarly tool such as the Oxford Concordance Program; but for the support of seminar and project work on literary texts *Personal Librarian* offers a valuable new tool which can be strongly recommended (Friedman *et al.*, 1990).

Personal Librarian requires an AT or above, with hard disk drive. Versions are also available for other operating systems and mainframes, and a Macintosh version is expected shortly. In the UK *Personal Librarian* is available from Systematic Upgrade, 58–60 Edward Road, New Barnet, Herts., EN4 8AZ. Telephone 01-449 9699. Price for a single user: £695. In the USA: Personal Library Software, 15215 Shady Grove Road, Suite 204, Rockville, Maryland 20850. Telephone 301-926-1402. Price: $895.

References

Friedman, E. A., McClellan, J. E., and Shapiro A. (1990). 'Automated Text Retrieval in Humanities Courses'. In D. S. Miall (ed.), *Humanities and the Computer: New Directions*, (ed.) 103–12. Oxford: Oxford University Press. Contains an account of a course based on *Personal Librarian*.

Koll, M. B., Noreault, T., and McGill, M. J. (1984). 'Enhanced Retrieval Techniques on a Microcomputer', *National Online Meeting: Proceedings—1984*, 165–70. Medford, NJ: Learned Information Inc.

Noreault, T., Koll, M., and McGill, M. J. (1977). 'Automatic Ranked Output from Boolean Searches in SIRE', *Journal of the American Society for Information Science*, **28**, 333–9.