6 CONCLUSIONS & RECOMMENDATIONS

6.1 Performance of the experimental search systems

The results of the evaluation experiment given in Chapter 5 show clearly that almost all the subjects who used the query expansion (qe) system found it more helpful than the dumb system. The query expansion facility was felt to be an important factor in this perceived helpfulness. Most subjects also judged the qe system to be as easy or easier than the dumb system. There was less consensus among subjects who used the full system. About four-fifths of them felt that it was more helpful than the dumb system, but more than half judged the dumb system to be easier (Table 5.2). In short, the qe system was highly acceptable, while the full system was considered usable, but markedly less acceptable than the qe. If user acceptability is the main criterion, a system based on the qe but with the interaction improved to counter some of the criticisms quoted in 5.7 could scarcely fail to be an improvement on most existing end user reference retrieval systems.

When quantitative results are considered the difference between the qe and full systems is less clear cut. Certainly the qe is more efficient if efficiency is measured in terms of the amount of, or the time spent in, scanning screens of records. But the precision figures in Table 5.8 show that users of the full system retrieved lists of books which were in a real sense better than the lists obtained by users of the qe system.

Both query expansion and classification browsing appear beneficial enough to justify evaluation in live use. But interactional and perhaps also functional modifications are needed before this can be done. These are outlined in later sections.

6.2 Query expansion

We would probably not have implemented automatic query expansion if the feasibility studies mentioned in 1.6 had not given encouraging results. The experiment of Smeaton and Van Rijsbergen discussed in 2.3.1 showed that a variety of techniques of automatic query expansion had detrimental or neutral results at least with the queries and the indexing which they used. It has been reported that *semi*-automatic query expansion can be beneficial in interactive systems, in CITE (2.4.2) and in Porter's system (2.4.3) for example, although we are not aware of any quantitative evaluation.

It is perhaps a little surprising that the automatic query expansion function was so successful. It might have been expected that automatic query expansion using system-selected terms from sparsely indexed library catalogue records would be erratic and unreliable. However, the records at the top of the ranking list of those retrieved by query expansion proved to be a somewhat richer source of relevant records than those obtained from the Dewey sequence. The screens of query expansion records were also far more consistent than the classified sequences.

Several users complained that classified displays often showed records unrelated to their needs, but no user made such a complaint about query expansion. More than two-fifths of the records chosen by users of the qe system were retrieved using it. There were very few critical remarks from users about the quality of the lists of records retrieved, although a number did comment unfavourably on the way in which records they had already seen reappeared in the query expansion lists. It seems that the present implementation of automatic query expansion is functionally satisfactory enough to form the basis of a live installation.

There are two areas where further development may be needed for live use. The first is concerned with heuristics for deciding when to advise the user that it may be worth invoking an expansion search, and the second with the mechanics of selecting and rejecting records to signify their relevance or otherwise. The following two sections give no more than an indication of some of the things which need to be considered.

6.2.1 Towards heuristics for offering query expansion

In the experimental systems the query expansion option was almost always explicitly available whenever there had been a change in the relevance information since the last query expansion search. In live use many user needs would be satisfied without query expansion. Unnecessary prompts clutter the screen and render the system significantly more difficult to use, so it is important that messages should be limited to those which are likely to be useful at any particular time. Further, even when a query expansion search "succeeds" the results are not always useful. Some users of the qe system, after initial success with a query expansion search, tended to use the facility after almost every selection of a record. These superimposed searches generally give record lists which are very similar to each other, loading the system unnecessarily and possibly confusing the user. Theoretically, a more methodical approach should give better results, although we were unable to obtain convincing evidence that this was true in practice. The sort of procedure envisaged is one where the user selects or rejects a good proportion of the readily available records at each stage, before invoking query expansion. This provides the maximum amount of information to the term extraction, weighting and selection processes.

The above considerations suggest that query expansion should not be offered unless the following conditions hold:

((1) several records have been judged relevant since the last query expansion

OR

(2) query expansion has not been tried, and the search appears rather unsuccessful or the user appears to be looking for more records)

AND

((3) the list of terms extracted from relevant records looks promising

OR

(4) a trial search produces at least one record of good weight)

It would be necessary to experiment with criteria for the truth of (1) -

(4) above. (1) and (2) are concerned with the need for query expansion (or some other method of seeking additional records) and (3) and (4) with the likelihood of its producing a fruitful list of records. Condition (1) is fairly straightforward. Indications for (2) include the following:

few titles retrieved at all
titles retrieved but few looked at in full
no records selected from first screen
repetition of same search, or searches with several terms in common.

Questioning the user may be appropriate in cases of doubt. It is not obvious what the criteria for (3) might be; it is always more straightforward to do a trial search (4), but this does pose an additional load on the system which should be avoided if possible.

6.2.2 The collection of relevance information

In the experimental systems users were motivated to look at records in full, one at a time, and to answer a relevance question, because their task was to compile a printed list of records. The relevance question was not recognized as such, but was rather taken as a question about whether the record should enter the print list or not. We feel that this way of obtaining relevance information would sometimes be unduly intrusive in live use of, for example, a library catalogue. Many user needs are satisfied by finding one or two relevant items, and many searches will readily be satisfied without query expansion. In any case it is not always practical to offer a facility for producing printouts of the selected records. A more acceptable approach may be to try to obtain information rather in the manner in which it is done in CITE (2.4.2), by asking a general question, which the user is not compelled to answer, at the foot of each screen of brief records. CITE shows records of intermediate length on a scrolling screen, and periodically gives users the opportunity of indicating which ones are relevant. Instead of demanding a reply to a relevance question, if the system detects a need it could prompt:

If you indicate which of these titles look useful the computer may be able to find other items which are similar to them.

Please type the numbers of any of these titles which look useful:

The whole procedure is very much simpler if there is only one level of record display, a level which is full enough to include subject headings. This was tried in Okapi '86. A few interviewed users complained that it was tedious seeing only one record at a time, although others said they preferred it. It makes the system easier to operate, but seeing only full records sometimes feels rather like using a magnifying glass; one would often prefer more of a bird's eye view. One of the subjects in the present experiment remarked that it was often possible to accept or reject records from the brief display alone. Many systems have a number of display formats, but remain in one default or chosen format until a different one is chosen. One possibility is to reverse the roles of full and brief records by using full as the default with the option of a display of brief records.

So far we have been assuming that interaction is through a conventional vdu and keyboard. The problems are certainly altered and in some ways reduced if the system can be offered on a bit-mapped screen with selection by mouse or tracker ball. During the development of the present systems Julie Porteous constructed a mock-up of a version of the dumb system running on a Sun workstation with mouse interaction. Selection of a record from the brief display leads to the rapid appearance of the full record in its own non-overlapping window. Buttons or a menu appear, offering "Select", "Reject" or "Continue". The full record remains displayed until the user selects another record or moves to a new screen of brief records. There are further possibilities in systems with graphical interaction. It may, for example, be feasible to give users who can benefit from it the ability to select or reject query expansion terms - this is semi-automatic rather than automatic query expansion (see 1.2 for definition). Unfortunately it is not yet possible to offer a system with graphical interaction for general use. Most library catalogues must be instantly usable, and there are many users who would have no idea what to do with a mouse. Probably the equipment is still beyond the reach of many institutions, although this may not be true for long. Also remote networked access to such systems is still not a practical proposition.

6.2.3 Computational implications of query expansion

Query expansion using terms extracted from relevant records needs considerable additional computational resources. In comparison with a straight "best match" system there will be more searches, and most of the searches will involve a substantially greater number of terms than the original search on the query terms. Hence in a practical implementation it may be important to limit the number of query expansion searches and/or to reduce the number of terms involved as much as possible. These limits were mentioned in 3.9.1. The rest of this section is a little technical and should be omitted by readers who are not interested in implementation details.

It is difficult to give estimates of the cpu and disk resources required for searches, because much will depend on hardware - particularly on the amount of memory available to the process, on file structures, operating system and application program design. A search is made up of one or more term lookups (to determine the number and location of the postings for each term), followed by a merge of the postings lists for the terms, and finally a sorting operation to place the output postings in decreasing weight order. These three operations do not necessarily take place consecutively; for a query expansion search the terms will already have been looked up during, or before, the process of assigning weights to them. At the other end of the process, the sort need not be done at all unless the user decides to look at the retrieved records. In fact in previous versions of Okapi (and some other systems of this type), the postings are not sorted at all. If the user chooses to see records the record display procedure repeatedly reads the list of postings and selects the one of highest weight remaining, thus in effect sorting by selection. This is not an efficient process, but in general the user will only wish to see a small proportion of the records which have been retrieved.

Hence we are primarily concerned with the merge itself. If the total number of postings for all the terms is P, and there are T terms, the number of disk accesses will be at least max(T, P/A), where A is the greatest number of postings which can be read in one disk access. The constant A depends on the size of the postings and on hardware and operating system characteristics. For our systems it is 1024, because the disk read unit is 4096 bytes and the postings are 4 bytes. There are of course at least T accesses, because a separate disk access is almost always required to read the postings for each term, however few there may be. (In passing, it should be noted that the number of postings for a term should be recorded in the index. Some search systems do not do this, so that finding the number of postings for a term can only be done by counting. Since this number is needed to assign a weight to a term it has to be known before the merge starts. If it is not recorded in the index every posting has to be read twice, once during the calculation of term weights and again during the merge.) Assuming that we never process more terms than can be merged in a single pass the amount of cpu time required for the merge is about (B + Clog(T))P or (B + DT)P for some constants B, C and D. The log(T) or T factor represents the time taken in deciding which input stream of postings to consider next. Which formula applies depends on the design of the merge algorithm. If the number of terms is rarely more than about eight or twelve the linear design (second formula) is probably as good as the logarithmic one, which is more complicated and has higher overheads. Our experimental systems used a linear design. The design of such merge algorithms is given in computer programming textbooks such as Horowitz and Sahni [HORO77, Section 8.2.1].

There is a simpler and faster merge, which takes time directly proportional to the total number of postings and is independent of the number of terms. It needs enough memory to store a weight for every record in the database. The entire array is initialized to zero, then all the postings are read, and the weight associated with each is added to the weight (initially zero) stored in the corresponding array element. This is the method used by Willett and others [HEND86b]. For large databases it needs several megabytes of internal memory for each active search process. Unlike other merge algorithms it has the potential advantage of being suited to parallel processing.

6.3 Classification browsing

Classification browsing was less good than query expansion as a source of relevant records. This surprised us, but perhaps it should not have done so, because query expansion searches are almost always based on more information - more index terms and usually more than a single relevant record. Many of the query expansion terms may be poor content indicators but this seems to be outweighed by other factors.

Nevertheless, more than two-fifths of the records chosen by users of the full system were chosen from the shelflist sequence (Table 5.4), so there is no doubt that classification browsing is often a valid source of additional relevant records. It is a facility which is computationally very light and easy to implement, at least in the simple way which we used. It should certainly be available as a retrieval tool if only because a great deal of intellectual effort is invested in the classification of bibliographic items, an effort which is in most cases only produces a shelving device.

6.3.1 Why classification browsing was relatively unsuccessful

For a large majority of the questions used in the experiment (Appendix 4) there is no single Dewey number which closely represents the topic. Readers who need to be convinced of this should try looking up the topics in the Dewey Relative Index [DEWE79, vol 3]. For most of the topics there is more than one appropriate number, and many of these numbers represent broad concepts which have some overlap with the sought topic. Almost any of the questions on the topic sheets will serve as an example. For "Role of the mother in the development of the child" most of the chosen records were found in four places within 155.4 (child psychology), at 306.87 (intrafamily relationships), several places in 362 (social welfare) and at 401.9 (psycholinguistics). Some analysis was done of the spread of class numbers among the records chosen by subjects in the experiment. This is difficult to summarize because of the wide variation in the number of people who searched for each topic (and hence the number of records found), but the overall picture for the 24 topics where more than ten distinct records were chosen is given in Fig 6.1. Half of the topics had three or four main class numbers, and there was only one topic where most of the records were brought together by the classification.

Fig 6.1 Distribution of the number of Dewey classes for 24 topics

```
                (8)
                 *
                 *
                 *
                 *
                 *
                 *
                 *
                 *     (4)
                 *      *
         (3)     *      *                   (3)
          *      *      *                    *
          *      *      *     (2)    (2)     *
          *      *      *      *      *      *
  (1)     *      *      *      *      *      *     (1)
   *      *      *      *      *      *      *      *
   *      *      *      *      *      *      *      *
---1------2------3------4------5------6------7------8---
              Number of Dewey classes
```

Other cases fail because of a lack of specificity in the classification. Here there are often too many items classified at one code for scanning and selection to be a practical retrieval mechanism. This was particularly noticeable in the area of computer science, although this has since been remedied in the Dewey classification. Conversely, there are cases where the classification is too specific. This rarely, if ever, happened with the topics used in our experiment, but it is likely that it would sometimes happen in live use. There is also the effect of collection size and range. Presumably it is reasonable to use a broader classification if the collection is small and non-specialized than if it is large or specialized. If there are very few items on most topics then it is not asking too much of the user, and recall may be quite good, if the user is shown every item classified in the same general area as a known relevant work.

6.3.2 Classification browsing in a live system

Our full system was designed to encourage use of classification browsing, and we would not have attempted such a crude implementation as that shown in Fig 3.12 in a system intended for live use. Several commercially available systems (BLCMP for example) offer a very similar facility to ours, sometimes in an even less developed form. In the BLCMP system at Warwick University, available on JANET, it does not even seem to be possible to return to the set of records retrieved by the original search after looking at a classified sequence. It is worth considering better ways of implementing classification browsing than we or BLCMP have used.

As with query expansion, classification browsing ought only be offered if it appears to be needed and if it seems likely to be useful. The evidence for need is similar to that which would be used for deciding whether there is a need to offer query expansion. Likelihood of usefulness is at least as difficult to estimate. As with query expansion, it can be done by silently retrieving some records and checking whether they would score reasonably highly if they had been retrieved by a query expansion search. This method can only be used in a system which also incorporated query expansion functions. Other clues, perhaps more practical, include the existence of some, but not too many, other records classified identically to the pivot record, or, in a system which collects relevance information, the fact that several selected records have the same classification. The latter is dependent on some records having been chosen relevant. It makes for a more responsive system if the program maintains a record of the class numbers at which classified display has already been seen during the current search. Subsequent requests for identical classified displays may then evoke a message to the effect that "You have seen this portion of the shelflist already". This does not of course inhibit the availability of the option, but repetition of searches from the transaction logs taken during the experiment reveals that users did quite frequently and apparently unintentionally repeat classified displays.

The display of records in classified sequence also needs refinement. It seemed logical to us to display the pivot record in the centre of the first screen, because this should suggest to the user that it is equally valid to browse in either direction. However, at least one full system user asked for the classification display to take some account of the number of items classified at a given code (5.5.3). This is most easily done if the initial classified display always includes the first record classified at the given code. Probably the last record at the previous code should also be shown. Commands to skip to the next and previous code will be useful when there is a large number of records at the current code. With a hierarchical classification such as Dewey there will be occasions where the ability to move up and down the hierarchy would be more useful than a linear browse. The DOC online catalogue developed by the Dewey Decimal Online Project workers [MARK86] allowed this, although it was the classification schedules which were displayed in the first instance rather than records. The results were not encouraging enough to suggest that work on hierarchical browsing should be a high priority.

6.3.3 A note on synthetic Dewey numbers

One of the objects given in the project proposal was to look into the feasibility of making use of the information given in those Dewey numbers which contain synthetic elements added from the tables of standard subdivisions, areas etc, and from the special instructions at many places in the schedules [DEWE79]. The intention was that synthetic numbers should be decomposed and used both when the database is indexed and when it is searched. Additions from the tables would be treated as search keys in their own right.

Because the informal testing at an early stage in the project suggested that Dewey numbers were likely to prove less useful in query expansion than subject and title words, no work was done on the decomposition of synthetic numbers. Leaving aside the question of usefulness, it appears that many synthetic numbers cannot be algorithmically decomposed. It can of course be done by using a large look-up table containing much of the information given in the schedules. Wajenberg [WAJE83] proposed a scheme for the MARC coding of Dewey numbers to enable algorithmic decomposition. This has been frequently cited, particularly in the publications of the Dewey Decimal Online Project. If it were adopted, which does not seem at all probable, Wajenberg's scheme would undoubtedly open the way to making more effective online use of the Dewey classification.

6.4 Further work

6.4.1 When and why does automatic query expansion work?

Unlike classification browsing, automatic query expansion is not dependent on a single key, nor in general on a single relevant record. An effect of the use of multiple keys is that bad ones tend to cancel each other out because they tend not to co-occur, but good ones reinforce each other because they do co-occur. Bad keys are ones like "natural" and "selection" extracted from a record on "Biochemical insect control" in the search for "Insecticides and the environment" (Fig 3.12). If used alone these terms find 13 books on evolution and no more books on the sought topic. But in combination with a number of good terms they are completely harmless. Nevertheless, repeating searches from the experiment shows that the results of query expansion searches vary a great deal in quality. We tried to make a classification of cases, because this could help in the development of heuristics for advising the user whether to try query expansion, and it might lead to some guidelines for weighting functions and term selection procedures. For a specific search it is usually rather easy to see why it behaves as it does, but we were not able to arrive at a fruitful classification, mainly because of the number and variability of the factors involved. These include the nature of the user's query - its length and the appropriateness of the user's terminology, the number of records assessed and the user's relevance judgments, the variation in the extent and nature of the indexing, and of course the coverage of the database. Appendix 8 is a partial transcript of a real search which illustrates some of the points about the unpredictability of query expansion.

6.4.2 Automatic query expansion with non-catalogue databases

The Okapi experiment described in Chapters 4 and 5 was done on a general catalogue database and under somewhat artificial conditions. Before it can be concluded that automatic query expansion is of general value in end user reference retrieval it must be evaluated in live use with different types of database and a range of user requirements. It is hoped that at least two Okapi installations will be set up at City University, London. One will access the library catalogue and another will access one of the abstracting and indexing databases. These will collect transaction log data from live use, and can be made available for controlled experiments. In view of the results obtained by Smeaton and Van Rijsbergen [SMEA83], it is important to find out how automatic query expansion performs on a more deeply indexed database.

6.4.3 Catalogue systems with enhanced records

In a previous report we concluded by urging "proper analytical indexing using contents pages and added free language descriptors". It is expected that work will start soon at Bath University on an Okapi system accessing monograph records which have been enhanced with contents and other subject-descriptive material derived from publishers' information. This will provide the opportunity for experimenting with ways of providing end-user access to material which, as regards indexable content, lies somewhere between bibliographic reference records and full text. It has not yet been decided whether query expansion will be tried in this system. It may be more important to work on the indexing of this material, and in particular on methods of term weighting and combination, as well as on the display of information.

6.4.4 Towards highly interactive systems: user monitoring as a tool for guiding interaction

It must by now be well known that if a functionally rich system is made available to general users very few will be able to use the full power of the system. Online and offline help and explanation facilities are little used. Simple systems with all their limited facilities up front work well (example: the "dumb" Okapi). Elaborate systems either swamp users with indigestible information or are adequately used only by a small number of experts. It is noteworthy, but not surprising, that the "full" Okapi system was seen by many users as being something of an effort to use. It is a comparatively elaborate system.

Heuristics for advising on the use of query expansion were briefly discussed above in 6.2.1. These involve making inferences from the behaviour of both the system and the user. To advance further in the development of retrieval systems for end users it will be necessary for systems to gather both short and long term information about individual users and uses, so that they can modify their interaction styles and modes, and their functionality, to suit the current need. There has been much discussion of such adaptive systems, and some ambitious proposals. One of the better known examples is the I³R system described by Croft and Thompson [CROF87]. Belkin and others [BELK87b] have provided some groundwork for the development of retrieval systems which perform many of the functions of the human search professional or intermediary.

There are two directions towards a groundwork for the design of such systems on which some work is urgently needed. The first is that of determining which characteristics of users and uses would enable a system to modify its functionality and interaction fruitfully. This is mainly a piece of desk work, requiring a thorough knowledge of information retrieval system design, and a survey of the literature. The second area is, in the context of end user reference retrieval, to investigate methods of gathering reliable user and use information in ways which are acceptably unobtrusive. It involves gathering user and use information, from live use of a system, and testing it for reliability and accuracy. It appears that there may be three types of clue from which useful information may be deduced. The first type consists of those which may be obtained from measurement of aspects of user-system interaction such as timings, the use of facilities and the occurrence of "errors" or inappropriate actions. All these clues are highly system-dependent, but it may be possible to work towards some fairly general principles. The second type are deductions from observations from the user's behaviour with regard to the search itself. For example, in a system which gathers relevance information, if a user performs a search which retrieves some documents, but does not choose any of them as relevant, this is probably an indication either that the search was inappropriately formulated or that there is nothing relevant in the database. Again, a user who makes identical or similar searches several times in succession may be seeking an exhaustive search. This type of clue is much less system-dependent than the first. Finally, there is the gathering and confirmation of information by explicit questioning of the user.

6.5 Concluding remarks

The primary object of this project was to work towards the production of a document retrieval system for end users which incorporates automatic query expansion facilities. In particular, we had to find a satisfactory way of implementing the query expansion function (source of search terms, term weighting, cutoff rules), and of providing interaction which enables most users to make good use of query expansion. The results of the experiment show that a system using a "best match" original search technique and with automatic query expansion available on request using terms extracted from records chosen by the user, is promising enough to warrant testing and tuning in live use.

A secondary object involved an implementation and evaluation of the related facility of using a single chosen record as a pivot leading to browsing displays of records in classified sequence. This feature was also found to be useful, although less so than automatic query expansion. However, class browsing is cheaper and easier to provide than query expansion, and it is strange that there are very few implementations among existing systems. It is also the only way of retrieving inadequately indexed records.

Query expansion is more universally applicable than class browsing, because the effectiveness of the latter is heavily dependent on the classification scheme and the way it is used in the database, whereas the former can use almost anything in the records, and it has something of a self-correcting tendency. On the other hand it is difficult to find ways of giving the user enough control to allow the best use of this somewhat blunt instrument. It also makes quite large demands on computational resources, but this will doubtless be a factor of decreasing importance.

This project has shown that end user retrieval systems would gain considerably in effectiveness from the incorporation of both query expansion and classification browsing facilities. There remains much work to be done on the development of heuristics for guiding user-system interaction and the availability and behaviour of the facilities.